NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

COMPUTATION OF WEAPONS SYSTEMS EFFECTIVENESS

by

Yong Shern Neo

September 2013

Thesis Co-Advisors:	Morris Driels Christopher Adams

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704–0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2013	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE
COMPUTATION OF WEAPONS SYSTEMS EFFECTIVENESS

5. FUNDING NUMBERS

6. AUTHOR(S) Yong Shern Neo

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943–5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. I.R.B. Protocol number N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

The main objective of the thesis is to develop an unclassified MATLAB-based Weapons Systems Effectiveness program with user-friendly Excel-based Graphical User Interface to evaluate the effectiveness of Air-to-Surface (AS) and Surface-to-Surface (SS) weapons. The program allows users to compute the effectiveness of their weapons based on the initial release/firing condition for both AS and SS weapons. The effectiveness is determined by predicting the probability of success of damaging or incapacitating the target. The program also allows the calculation of the trajectory of unguided AS weapons and the computation of the weapon accuracy of AS weapons.

The main functionality provided by the program includes computing the weapon effectiveness of single release of weapons against unitary and area targets, stick deliveries, cluster munitions and projectiles for AS weapons, as well as direct and indirect SS weapons.

The entire program is modeled on the theory detailed in M. R. Driels' textbook, *Weaponeering Conventional Weapon System Effectiveness*, published in 2013 by the AIAA.

14. SUBJECT TERMS Weapons Effectiveness Program, Probability of Damage, Weapon Trajectory, Weapon Accuracy

15. NUMBER OF PAGES
175

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540–01–280–5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

COMPUTATION OF WEAPONS SYSTEMS EFFECTIVENESS

Yong Shern Neo
ME5, Singapore Arm Forces
B.Eng., Nanyang Technological University, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING SCIENCE
(MECHANICAL ENGINEERING)

from the

NAVAL POSTGRADUATE SCHOOL
September 2013

Author: Yong Shern Neo

Approved by: Morris Driels
Thesis Advisor

Christopher Adams
Co-Advisor

Knox T. Milsaps
Chair, Department of Mechanical
and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The main objective of the thesis is to develop an unclassified MATLAB-based Weapons Systems Effectiveness program with user-friendly Excel-based Graphical User Interface to evaluate the effectiveness of Air-to-Surface (AS) and Surface-to-Surface (SS) weapons. The program allows users to compute the effectiveness of their weapons based on the initial release/firing condition for both AS and SS weapons. The effectiveness is determined by predicting the probability of success of damaging or incapacitating the target. The program also allows the calculation of the trajectory of unguided AS weapons and the computation of the weapon accuracy of AS weapons.

The main functionality provided by the program includes computing the weapon effectiveness of single release of weapons against unitary and area targets, stick deliveries, cluster munitions and projectiles for AS weapons, as well as direct and indirect SS weapons.

The entire program is modeled on the theory detailed in M. R. Driels' textbook, *Weaponeering Conventional Weapon System Effectiveness*, published in 2013 by the AIAA.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AL	Lethal Area (m^2 or ft^2)
AS	Air-to-Surface
BOC	Bombing-on-Coordinates
CCIP	Continuous Computed Impact Point
CCRP	Continuous Computed Release Point
CEP	Circular Error Probable (m or ft)
DEP	Deflection Error Probable (m or ft)
DMPI	Desired Mean Point of Impact
$E(F_C)$	Expected Fractional Coverage
EFD	Expected Fractional Damage
FBAR	Direct Fire Against Personnel Program
FCC	Fire Control Computer
GC	Guidance and Control
GUI	Graphical User Interface
ICM	Improved Conventional Munitions
IV Distance	Intervalometer Distance Setting (m or ft)
IV Time	Intervalometer Time Setting (sec)
JMEM	Joint Munitions Effectiveness Manual
JWS	JMEM Weaponeering System
LA	Target Length (m or ft)
L_B	Enlarged Weapon Lethal Length with Ballistic Dispersion (m or ft)
LEP	Enlarged Weapon Lethal Length (m or ft)
LET	Length of Effective Target Area (m or ft)
LET'	Effective Target Length for Carleton Damage Function (m or ft)
LOS	Line-of-Sight
L_P	Pattern Length (m or ft)
L_S	Stick Length (m or ft)
L_{SP}	Single Round Pattern Length (m or ft)
L_V	Volley Length (m or ft)
L_{VP}	Volley Pattern Length (m or ft)

MAE	Mean Area of Effectiveness (m^2 or ft^2)
MAE_B	Mean Area of Effectiveness for Blast (m^2 or ft^2)
MAE_F	Mean Area of Effectiveness for Fragmentation (m^2 or ft^2)
n_b	Number of submunitions in dispenser
n_r	Number of Release Pulses
n_x	Expected number of submunitions that can damage the target given target cover
P_{CD}	Conditional Damage Probability
$P_{Incapacitated}$	Probability of Incapacitation
R_B	Bomb Fall Range (m or ft)
R_b	Submunition Reliability
REP	Range Error Probable (m or ft)
SR	Slant Range (m or ft)
SS	Surface-to-Surface
SSPD	Single Sortie Probability of Damage
TER	Triple Ejection Rack
TLE	Target Location Error (m or ft)
WA	Target Width (m or ft)
W_B	Enlarged Weapon Lethal Width with Ballistic Dispersion (m or ft)
WEP	Enlarged Weapon Lethal Width (m or ft)
WET	Width of Effective Target Area (m or ft)
WET'	Effective Target Width for Carleton Damage Function (m or ft)
W_P	Pattern Width (m or ft)
W_S	Stick Width (m or ft)
W_{SP}	Single Round Pattern Width (m or ft)
W_V	Volley Width (m or ft)
W_{VP}	Volley Pattern Width (m or ft)
σ_b	Ballistic Error Standard Deviation (m or ft)
σ_{bd}	Ballistic Error Standard Deviation in Deflection (m or ft)
σ_{br}	Ballistic Error Standard Deviation in Range (m or ft)

ACKNOWLEDGMENTS

I would like to thank a few people who have been instrumental to the completion of my thesis.

First and foremost, I would like to thank Prof. Driels for the invaluable guidance and patience he has provided throughout the duration of the thesis process, especially during the long discussions we had on the theoretical aspects of the weapon effectiveness program.

Second, I would like to thank Prof. Adams for his invaluable insights from his days flying onboard the F-14, which contributed to the development of the Graphical User Interface. More importantly, I am grateful to Prof. Adams for helping me find a thesis that I am passionate about.

Lastly but not the least important, I would like to thank my family, especially my wife, for their patience and understanding on the time and effort I spent on my thesis at the expense of valuable family time.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

The main objective of the thesis is to develop an unclassified MATLAB-based Weapons Systems Effectiveness program with user-friendly Excel-based Graphical User Interface (GUI) to evaluate the effectiveness of Air-to-Surface (AS) and Surface-to-Surface (SS) weapons. While there are existing programs that perform such effectiveness computations, such as the JMEM Weaponeering System (JWS), they are classified due to the data in the program, which limits the number of users that can use the program. The Weapon Systems Effectiveness program provides an unclassified solution by adopting the same methodology used in the existing program, while maintaining an unclassified database.

The program allows users to compute the effectiveness of their weapons based on the initial release/firing condition for both AS and SS weapons. The effectiveness is determined by predicting the probability of success of damaging or incapacitating the target. The program also calculates the trajectory of unguided AS weapons and then computes the weapon accuracy of AS weapons.

The key advantage of this program compared to existing programs is that it incorporates a high-fidelity trajectory model to compute impact condition of the weapons, while the impact conditions are strictly user-inputs in existing programs. Furthermore, this program is a “one-stop” program which allows the computation of the weapon effectiveness for both AS and SS weapons, while existing programs are stand-alone AS and SS programs.

The entire program is modeled on the theory detailed in M. R. Driels’ textbook, *Weaponeering Conventional Weapon System Effectiveness*, published in 2013 by the AIAA [1].

B. WEAPON EFFECTIVENESS PROGRAM OVERVIEW

The functionality provided by the proposed program is as follows:

1. Trajectory Calculation of unguided AS weapons: This function calculates the trajectory of unguided AS weapons given the initial release condition of the weapon. Two additional trajectory modules were created for stick release and cluster deliveries due to additional computation requirements.

2. Delivery accuracy of AS weapons: This function computes the delivery accuracy of AS weapons based on the accuracy of the sensors utilized for the release of the weapon in Continuous Computed Release Point (CCRP), Continuous Computed Impact Point (CCIP) and Bomb-on-Coordinates (BOC) bomb modes.

3. Weapons effectiveness computation of single weapon against unitary targets. This function computes the probability of damaging a unitary target using a single AS weapon. This includes the computation for both guided and unguided weapons.

4. Weapons effectiveness computation of single weapon against an area of targets. This function computes the probability of damaging an area of targets using a single AS weapon. This includes the computation for both guided and unguided weapons.

5. Weapons effectiveness computation of stick deliveries. This function computes the probability of damaging an area of targets, including a single large target, using a stick of AS weapons. This includes the computation for both guided and unguided weapons.

6. Weapons effectiveness computation of cluster weapons. This function computes the probability of damaging an area of targets using either a single cluster weapon or a stick of cluster weapons.

7. Weapons effectiveness computation of projectiles. This function computes the probability of damaging a target using projectiles such as aircraft gun rounds or rockets.

8. Weapons effectiveness computation of indirect fire SS weapons against area targets. This function computes the probability of damaging an area of targets using indirect fire such as artillery. The indirect fire weapon can be a unitary warhead or an ICM. The function is modeled after the existing Superquickie2 program.

9. Weapons effectiveness computation of indirect fire SS weapons against point targets. This function computes the probability of damaging a point target using indirect fire with a Unitary Warhead. The computation is done using Monte Carlo simulation.

10. Weapons effectiveness computation of direct SS weapons against dismounted infantry. This function computes the probability of incapacitating enemy troops with direct fire using Monte Carlo-based simulation of the existing Direct Fire Against Personnel Targets (FBAR) program.

C. OVERALL SYSTEM ARCHITECTURE

The overall system architecture for the weapons effectiveness program is illustrated in Figure 1. There are two main programs used for the weapons effectiveness program. Excel is used as the main GUI for users as it is more user-friendly, while MATLAB is used as the computation tool due to its computational capability.

All system inputs are entered in Excel. There are two Excel files used in the program. The first Excel file serves as the main GUI for users to enter their data, while the second Excel file is used by the MATLAB program to update the results. The Main GUI Excel file has separate worksheets for entering of the weapon release/firing conditions and constant data such as the weapon parameters and constant variables.

The main user interface module in MATLAB is used to retrieve the input data from the main GUI in Excel. The main user interface module will then call out the required function module depending on the functionality that the user selected. All the function models are programmed such that they can be separately called without the main user interface module. When the results are computed, the function models will return the results to the main user interface module. The main user interface module will then call out the required Write-to-Excel functions to update the results into the Excel Result File. The results will then be updated in the main GUI when the refresh button is selected.

When the selected function modules require input data from the trajectory modules, they will call the trajectory modules directly for the results. There are separate trajectory models used for single weapons, stick deliveries and cluster weapons.

The trajectory module and weapon accuracy module require input parameters such as the mass of the weapons or the error specification of onboard sensors. These parameters are provided by the main GUI Excel file via the Weapon Parameter Module, Sensor Variable Module and Constant Table Module.

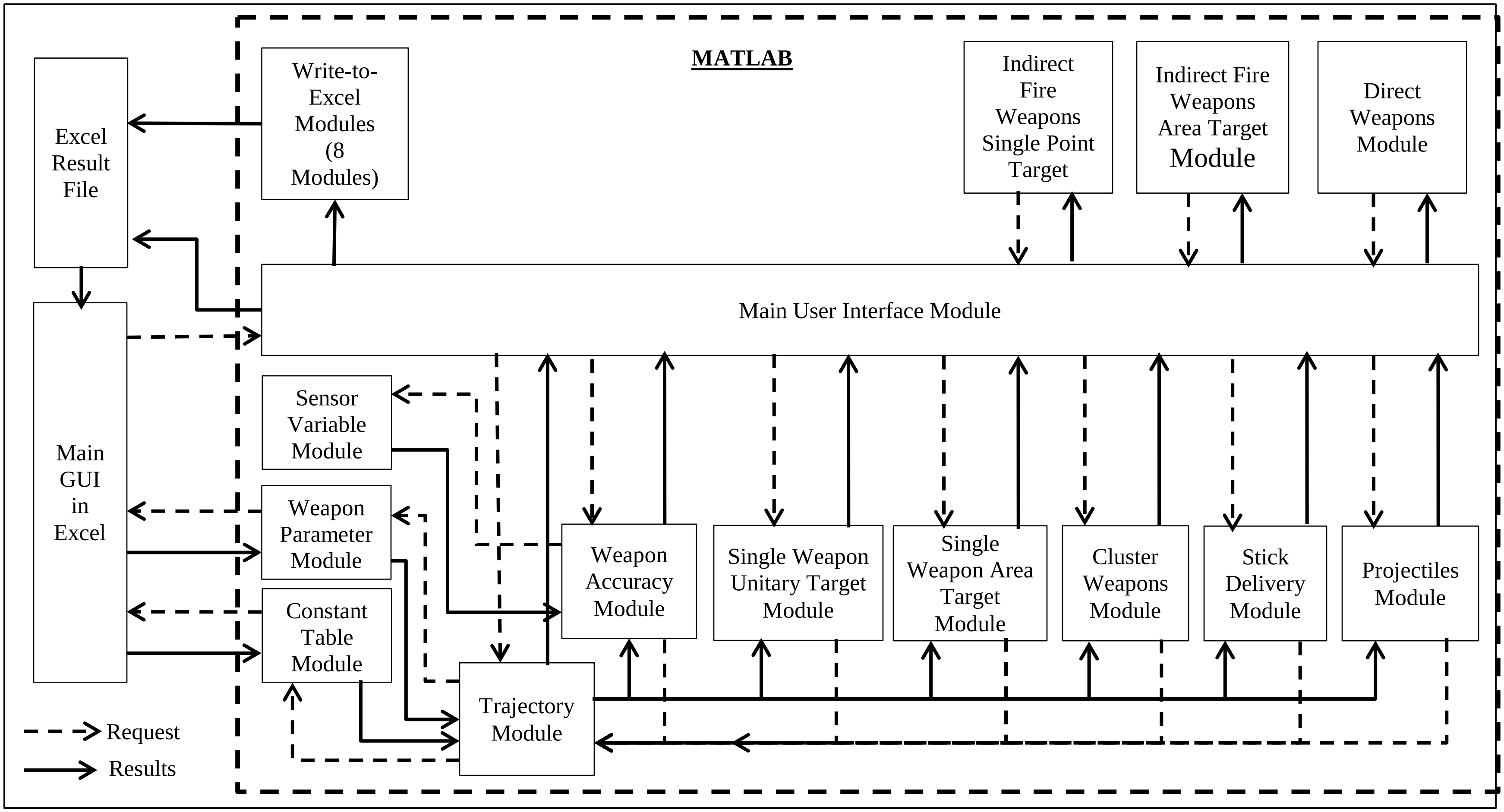


Figure 1. Overall System Architecture.

THIS PAGE INTENTIONALLY LEFT BLANK

II. WEAPON EFFECTIVENESS PROGRAM MAIN INTERFACES

A. BACKGROUND

This chapter covers the files used in the weapon effectiveness program that provides the interfaces between MATLAB and Excel. The Excel-based file includes the main GUI and Excel result file. The MATLAB-based file includes the main user interface, sensor variable module, weapon parameter module, constant table module and Write-to-Excel modules. The main purpose of the interface modules in MATLAB is to handle all the data exchanged between Excel and MATLAB. This eliminates the need for the core weapon effectiveness computation modules to interact with Excel, thus increasing their usability and allowing them to be used by other MATLAB programs.

B. MAIN GUI IN EXCEL

The main GUI, which is based in Excel and titled "Weaponeering_ Program_Main_User_Interface," provides four distinct functions, allowing:

1) Users to select the required functionality and input the necessary initial weapon release/fire conditions.

2) Users to view the results of the weapon effectiveness program.

3) Program administrators to input weapon parameters.

4) MATLAB to retrieve the input data required for its weapons effectiveness computation.

The main GUI consists of 10 essential worksheets as summarized in Table 1.

S/No	Worksheet Name	Purpose
1	Main User Input	A one-page worksheet that allows user to: 1) Select required functionality and enter desired release/firing condition for AS weapons. 2) Refresh the results using the refresh button in the worksheet. 3) View the overall result on the probability of damage of the weapon.
2	Air-to-Surface Detailed Results	This worksheet shows the detailed results for AS weapons, should user be interested on the intermediate results that lead to the final result for the probability of damage computation.
3	SS User Input	It serves the same function as the Main User Input worksheet with the exception that it is used for SS weapons delivery instead of AS weapons.
4	SS Detailed Results	It serves the same purpose as the Air-to-Surface Detailed Results worksheet with the exception that it is for SS weapons.
5	4 Sensor Variable Input	This worksheet allows the program administrator to input the sensor variable errors used for the weapon accuracy program.
6	5 Weapon Data Sheet	This worksheet allows the program administrator to input the weapon parameters such as the dimension and mass of the weapon, the accuracy of the weapon and the probability of kill parameters for guided weapons.
7	8 Constant Definitions	This worksheet allows the program administrator to input constant parameters such as gravitational acceleration constant.
8	9 Output to MATLAB	This worksheet converts the data selected/entered in the main user input worksheet for AS weapons into numerical values as input to the main user interface module in MATLAB.
9	9b Output to Matlab	This worksheet serves the same purpose as the previous worksheet for SS weapons.
10	Error Code	This worksheet provides the list of error code in the event that user did some wrong input selection.

Table 1. Essential worksheets in Main GUI Excel file.

C. EXCEL RESULT FILE

The Excel result file serves as an intermediate file in which MATLAB can update the results from the weapon effectiveness program. The results are then updated in the Main GUI when the refresh button is selected. The main purpose of the file is to allow the results to be updated on the Main GUI so the GUI can remain in use. However, it is not technically possible for MATLAB to write to an Excel file when the file is open. Hence the solution is to create an additional results file in Excel that serves as a proxy for MATLAB in which the results can be updated and from which the main GUI can retrieve those results. The Excel results file must not be opened during the simulation run as it will cause the weapon effectiveness program to generate an error. A screenshot of the main GUI is shown in Figure 2.

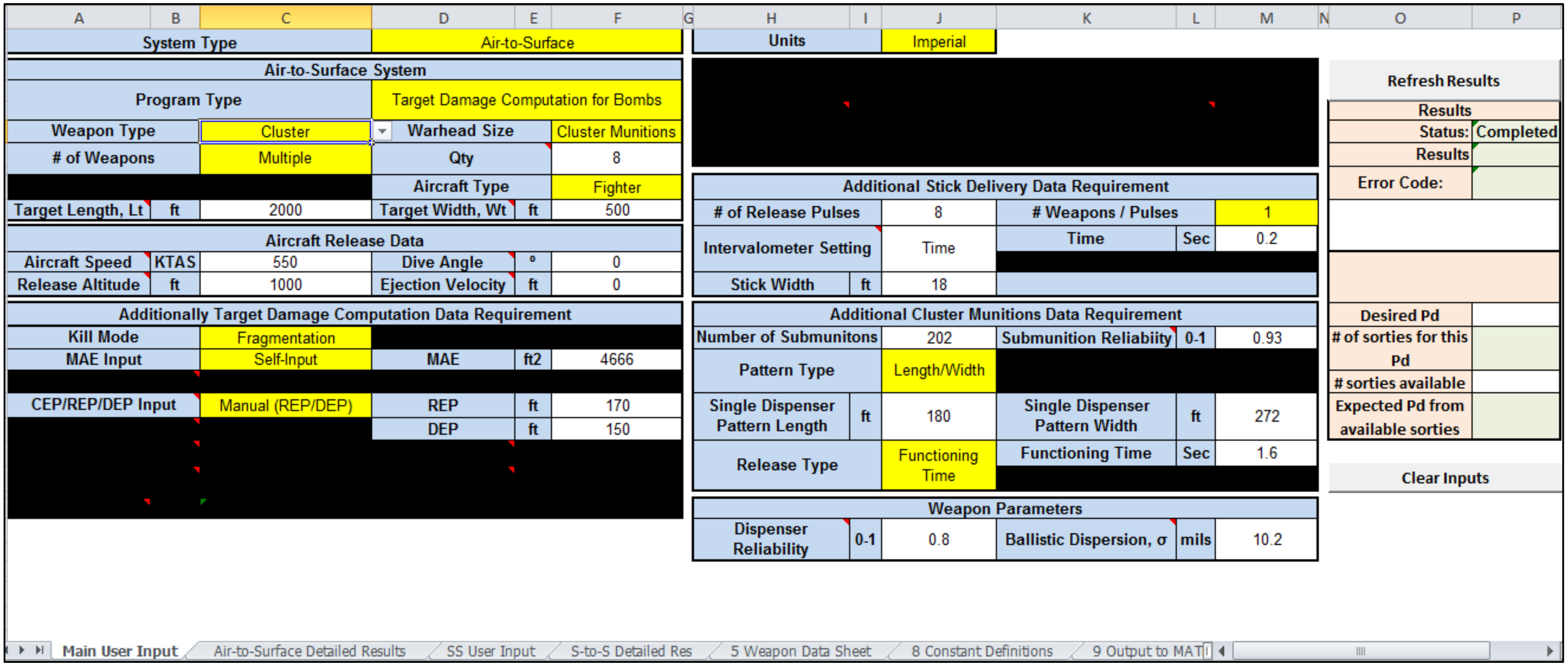


Figure 2. Screenshot of Main GUI in Excel.

D. MAIN USER INTERFACE MODULE

The main user interface module (Main_Excel_Interface.m) is the main interface module in MATLAB that runs the weapons effectiveness program. It retrieves the user input from the Main GUI in Excel and calls the appropriate function to perform the required weapons effectiveness computation. It receives the results from the respective modules and commands the results to be updated in the Excel file.

1. Selection of System

The module first decides on the type of system to run based on user selection as shown in Figure 3. The two types of systems available are the Air-to-Surface Program and the Surface-to-Surface Program.

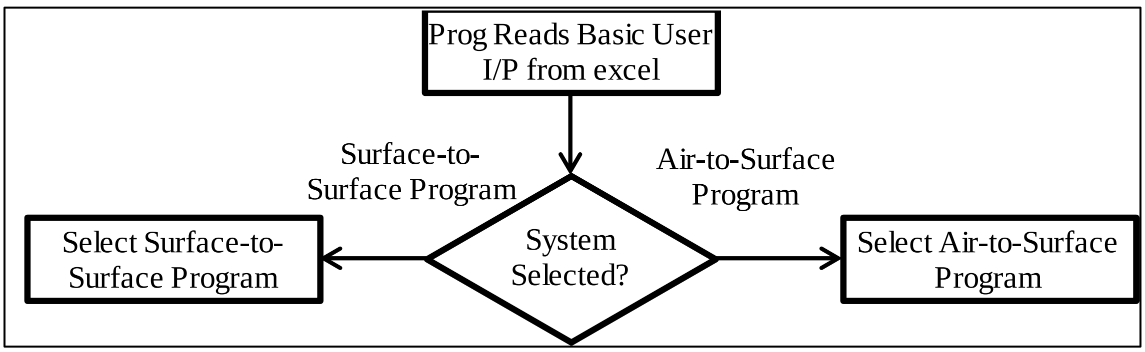


Figure 3. Selection of system type for Weapon Effectiveness Program.

2. Selection of Air-to-Surface Weapons Modules

Next, the required Air-to-Ground function is called based on the factors summarized in Table 2. The codes for the module are written based on the flow chart illustrated in Figure 4.

S/No	Factors Determining what Function to Call				Function Called
	Program Type	Weapon Type	Number of Weapons	Target Type	
1	Trajectory	N.A.	N.A.	N.A.	Trajectory Module
2	Weapon Accuracy	N.A.	N.A.	N.A.	Weapon Accuracy Module
3	Target Damage Computation for Rockets/Projectiles	N.A.	N.A.	N.A.	Rocket/Projectile Module
4	Target Damage Computation for Bombs	Cluster	N.A.	N.A.	Cluster Module
5	Target Damage Computation for Bombs	Others	Multiple	N.A.	Sticks Module
6	Target Damage Computation for Bombs	Others	Single	Point	Single Weapon Unitary Target Module
7	Target Damage Computation for Bombs	Others	Single	Area	Single Weapon Area Target Module

Table 2. Summary of factors determining AS function modules.

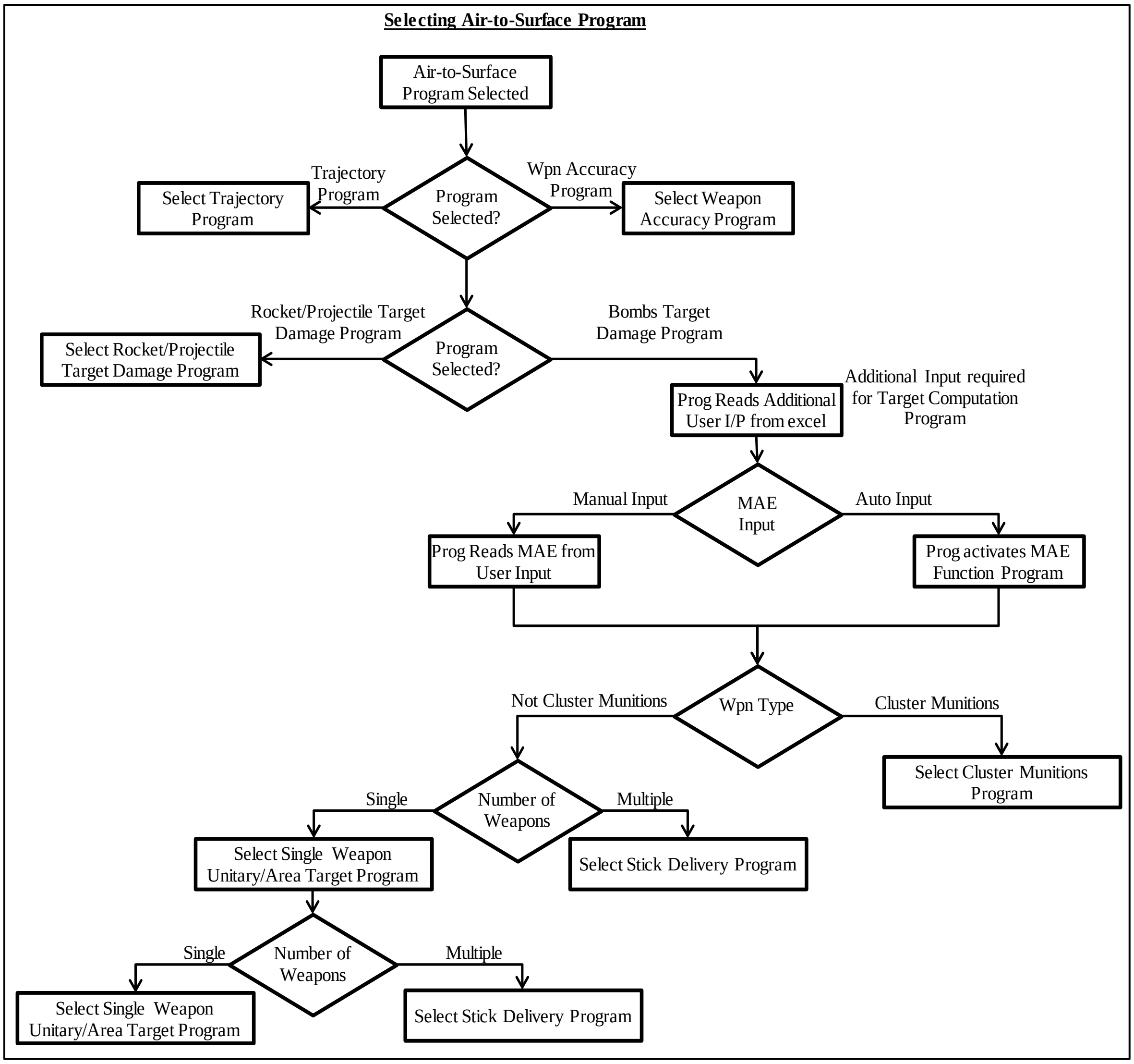


Figure 4. Flow diagram overview for Main Interface module.

3. Selection of Surface-to-Surface Weapons Modules

In the event that the Surface-to-Surface (SS) system is selected, the required Surface-to-Surface function is called based on the factors summarized in Table 3.

/No	Factors Determining What Function to Call		Function Called
	Program Type	**Target Type**	
1	Indirect Fire	Single	Indirect Monte Carlo Module
2	Indirect Fire	Area	Indirect Fire Weapons Module (Superquickie2)
3	Direct Fire	N.A.	Direct Fire Module (FBAR)

Table 3. Summary of factors determining SS function modules.

E. SENSOR VARIABLE MODULE

The sensor variable module acts as the main interface program for the weapon accuracy module to retrieve sensor error parameters from the Main GUI in Excel. It is modeled as a function with the input commands and resultant output shown in Table 4. When the module is called, it will retrieve the required data from Excel and send the results back to the program that called the function.

Input	Sensor_Variable_Model(Bomb_Mode, Air_Platform_Type, Targeting_Pod, Target_Position_Error_Type)
Output 1 **(CCRP Mode)**	Sensor_Variable_output = [velocity_x_error_sigma, velocity_y_error_sigma, velocity_z_error_sigma, roll_error_sigma, pitch_error_sigma, yaw_error_sigma, harp_angle_error_sigma, slant_range_error_sigma];
Output 2 **(CCIP Mode)**	Sensor_Variable_output = [velocity_x_error_sigma, velocity_y_error_sigma, velocity_z_error_sigma, roll_error_sigma, pitch_error_sigma, yaw_error_sigma, height_error_sigma];
Output 3 **(BOC Mode)**	Sensor_Variable_output = [velocity_x_error_sigma, velocity_y_error_sigma, velocity_z_error_sigma, roll_error_sigma, pitch_error_sigma, yaw_error_sigma, aircraft_position_error_x, aircraft_position_error_y, aircraft_position_error_z, Target_Position_Error_x, Target_Position_Error_y, Target_Position_Error_z];
Output 4 **(GPS Weapon)**	Sensor_Variable_output = [Target_Position_Error_x, Target_Position_Error_y, Target_Position_Error_z];

Table 4. Input and output command for sensor variable module.

F. WEAPON PARAMETER MODULE

The weapon parameter module variable module acts as the main interface program for the trajectory module to retrieve the weapon parameters from the Main GUI in Excel. It is modeled as a function with the input commands and resultant output shown in Table 5. When the module is called, it will retrieve the required data from Excel and send the results back to the program that called the function.

Input	`Weapon_Parameter_Model(Warhead_Size)`
Output 1	`weapon_parameter_output = [Bomb_W, Bomb_D, Cd];`
Output 2 (Cluster Weapon)	`weapon_parameter_output = [Bomb_W, Bomb_D, Cd, Sub_Bomb_W, Sub_Bomb_D, Sub_Cd];`

Table 5. Input and output command for weapon parameter module.

G. CONSTANT TABLE MODULE

The constant table Module variable module acts as the main interface program for the trajectory module to retrieve the constant parameters from the Main GUI in Excel. The constants parameters are being defined in Excel for the convenience of the system administrators when defining these constants. It is modeled as a function with the input commands and resultant output shown in Table 6. When the module is called, it will retrieve the required data from Excel and send the results back to the program that called the function.

Input	`constant_table( x )`
Output	`constant_table_output = xlsread(‘Weaponeering_Program_Main_User_Interface.xlsm’,’8 Constant Definitions’,’D2’);`

Table 6. Input and output command for constant table module.

H. WRITE-TO-EXCEL MODULES

The Write-to-Excel modules write the results from the Weapon Effectiveness Programs into the Excel results file. These modules are modeled as function files. They are called by the Main User Interface module after the respective Weapon Effectiveness Programs have completed their computation and returned their results to the Main User Interface module. There are a total of eight Write-to-Excel modules as follows:

1) Write_to_excel_Single_Target_Unitary_Weapon_Results
2) Write_to_excel_Single_Weapon_Area_Target_Results
3) Write_to_excel_Stick_Delivery_Results
4) Write_to_excel_Cluster_Results
5) Write_to_excel_Rockets_Projectiles
6) Write_to_excel_Trajectory_Results
7) Write_to_excel_Weapon_Accuracy_Results
8) Write_to_excel_Superquickie_Results

THIS PAGE INTENTIONALLY LEFT BLANK

III. WEAPON TRAJECTORY

A. BACKGROUND

The weapon trajectory module provides information on the trajectory of AS weapons based on the release profile of the weapon such as release altitude, speed and dive angle of the aircraft. From this module, the operator is able to generate the trajectory of the weapon and obtain parameters such as the impact velocity and impact angle of the weapon when it detonates. Together with the ballistic partial, such parameters can be used as the inputs for weapon accuracy and target damage computations. The ballistic partials can be viewed as changes in the ground range when there are perturbations in one of its independent variables such as altitude and velocity. It is used as one of the variables required to compute the error from unguided AS weapons as part of the weapon accuracy module covered subsequently.

While there are three ways of computing the weapons trajectory, only two of the trajectory programs will be developed; the zero-drag point-mass trajectory module and the high-fidelity trajectory module as illustrated in Figure 5. The high-fidelity trajectory program is used to compute the terminal parameters such as impact angle and impact velocity, as it provides the most accurate estimates. The zero-drag module is used to compute the ballistic partial as it is able to provide an acceptable level of accuracy for the computation of the ballistic partials at a faster speed compared to the high fidelity trajectory module. These ballistic partials are required as inputs for the weapon accuracy program.

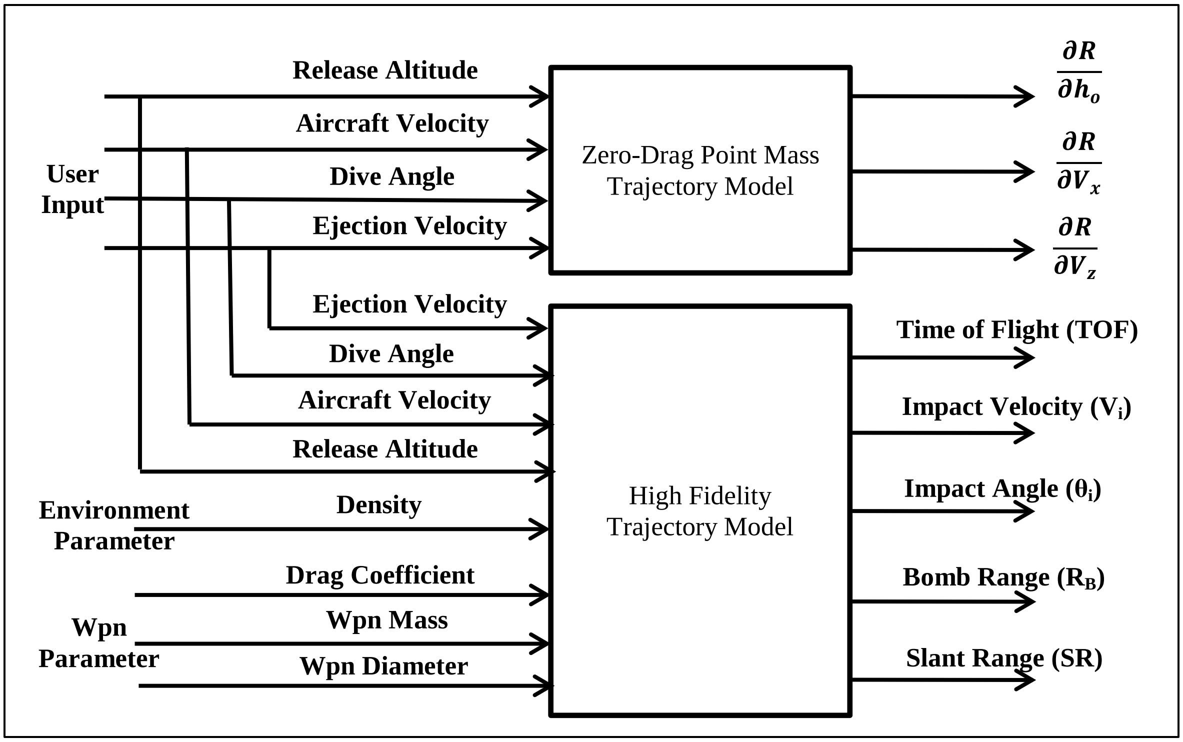


Figure 5. Trajectory modules input and output.

This thesis will not look into the trajectories of unguided surface-to-surface munitions, as the trajectory of such munitions is more complex to compute than that of air-to-surface munitions, as these weapons pass through the supersonic, transonic and subsonic regime. The trajectory of guided munitions is also not addressed as the trajectory of these weapons can be affected by external factors such as preset impact conditions and external lasing systems used to guide the weapon.

The materials in this chapter are taken from Chapter 3 of [1]. Additional theoretical information can be found in the same source.

B. ZERO-DRAG, POINT MASS TRAJECTORY MODULE

The zero-drag, point mass trajectory module computes the impact condition of the weapon based on the assumption that the weapon experiences zero drag and is represented by a dimensionless point in space. The inputs required by the module are summarized in Table 7.

S/No	Input Parameter	Input Method	Description
1	Release Altitude	User Input	The altitude of the aircraft where the weapon is released with respect to target altitude.
2	Aircraft Speed	User Input	The speed of the aircraft when the weapon is released.
3	Dive Angle	User Input	The dive angle of the aircraft when the weapon is released.
4	Ejection Velocity	User Input	The velocity in which the weapon is being ejected out of the aircraft by release cartridges, such as ARD-446.

Table 7. Zero-drag trajectory module input parameters

With the aircraft velocity, aircraft dive angle and ejection velocity specified, we are able to compute the weapon velocities at release using the formulas below. The various components of the weapon velocities are depicted in Figure 6.

$$V_{Ox} = V_a cos\theta - V_e sin\theta \tag{3.1}$$

$$V_{Oz} = V_a sin\theta + V_e cos\theta \tag{3.2}$$

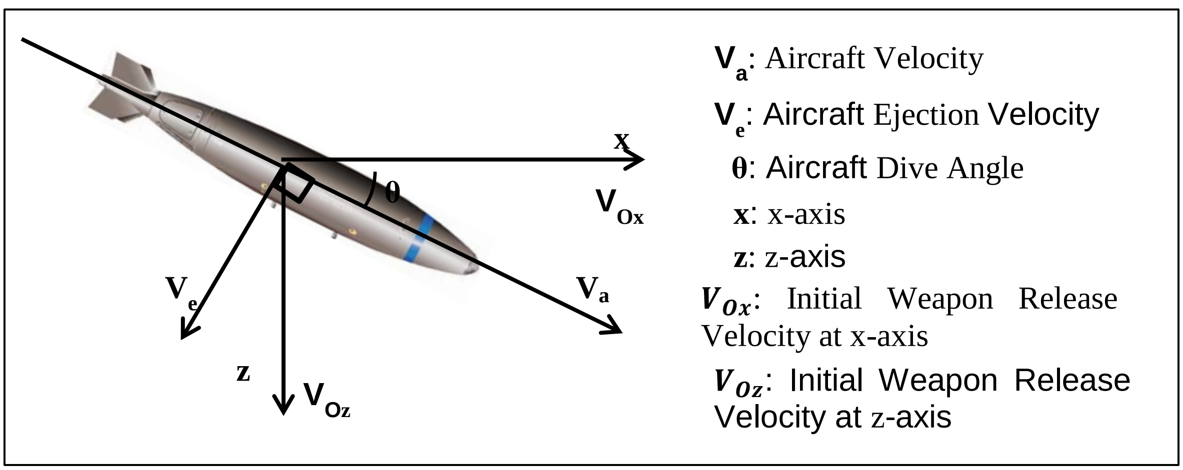


Figure 6. Weapon velocities at release.

With the initial release velocities determined, we are able to calculate the impact conditions of the weapon as follows:

$$Time\ of\ Flight, TOF = \frac{-V_{Oz}+\sqrt{V_{Oz}{}^2+2gh}}{g} \tag{3.3}$$

where:

g = 9.80665 m/s^2 or 32.174 ft/s^2

h = aircraft release altitude

$$\text{Impact Velocity (x} - \text{axis)}, \mathrm{V_{ix}} = \mathrm{V_{Ox}} \tag{3.4}$$

$$\text{Impact Velocity (z} - \text{axis)}, \mathrm{V_{iz}} = \mathrm{V_{Oz}} + (\mathrm{g} * \ \mathrm{TOF}) \tag{3.5}$$

$$\text{Impact Velocity}, \mathrm{V_i} = \sqrt{\mathrm{V_{ix}}{}^2 + \mathrm{V_{iz}}{}^2} \tag{3.6}$$

$$\text{Impact Angle}, \theta_\mathrm{i} = \tan^{-1}\frac{\mathrm{V_{iz}}}{\mathrm{V_{ix}}} \tag{3.7}$$

$$\text{Bomb Range}, \mathrm{R_B} = \ \mathrm{V_{ix}} \ * \mathrm{TOF} \tag{3.8}$$

$$\text{Height of Weapon Travelled} = \mathrm{h} = (\mathrm{TOF} * \mathrm{V_{iz}}) + (0.5 * \mathrm{g} * \mathrm{TOF}^2) \tag{3.9}$$

$$\text{Slant Range}, \mathrm{SR} = \sqrt{\ \mathrm{R_B}{}^2 \ * \mathrm{h}^2} \tag{3.10}$$

With the impact conditions computed based on the specified weapon release condition, we are able to compute the ballistic partials to examine the effects that varying h, V_{Ox} and V_{Oz} have on R_B using the following equations:

$$\frac{\partial \mathrm{R_B}}{\partial \mathrm{h}} = \frac{\text{New } \mathrm{R_B} - \text{Old } \mathrm{R_B}}{\text{New h} - \text{Old h}} \tag{3.11}$$

$$\frac{\partial \mathrm{R_B}}{\partial \mathrm{V_{Ox}}} = \frac{\text{New } \mathrm{R_B} - \text{Old } \mathrm{R_B}}{\text{New } \mathrm{V_{Ox}} - \text{Old } \mathrm{V_{Ox}}} \tag{3.12}$$

$$\frac{\partial \mathrm{R_B}}{\partial \mathrm{V_{Oz}}} = \frac{\text{New } \mathrm{R_B} - \text{Old } \mathrm{R_B}}{\text{New } \mathrm{V_{Oz}} - \text{Old } \mathrm{V_{Oz}}} \tag{3.13}$$

The old values are the values computed based on the initial specified weapon release conditions. Hence $\frac{\partial R}{\partial h}$ is computed by making a small increment in h and re-computing the new R_B using the equations above while keeping the rest of the variables unchanged from the initial values. Likewise, $\frac{\partial R_B}{\partial V_{Ox}}$ and $\frac{\partial R_B}{\partial V_{Oz}}$ are computed by varying V_{Ox} and V_{Oz}, respectively and re-computing the new R_B while keeping the rest of the variables unchanged from the initial values. These ballistic partial outputs will be used subsequently as inputs for the weapon accuracy program.

1. Zero-Drag Trajectory Module in MATLAB

The zero-drag trajectory module is modeled as a function. The input commands and the associated outputs for the zero-drag trajectory module are depicted in Table 8.

Input	`Zero_Drag_Trajectory_Model(release_altitude, dive_angle, ac_velocity, ejection_velocity )`
Output	`Zero_Drag_Trajectory_output = [TOF, impact_velocity_x, impact_velocity_z, Impact_Angle, Impact_Velocity, SR, Rb,Partial_Derivative_Rb_h,Partial_Derivative_Rb_Vx,Partial_Derivative_Rb_Vz ]`

Table 8. Input and output commands for zero-drag trajectory module.

The zero-drag trajectory module is modelled using the equations explained, with a flowchart of the program depicted in Figure 7.

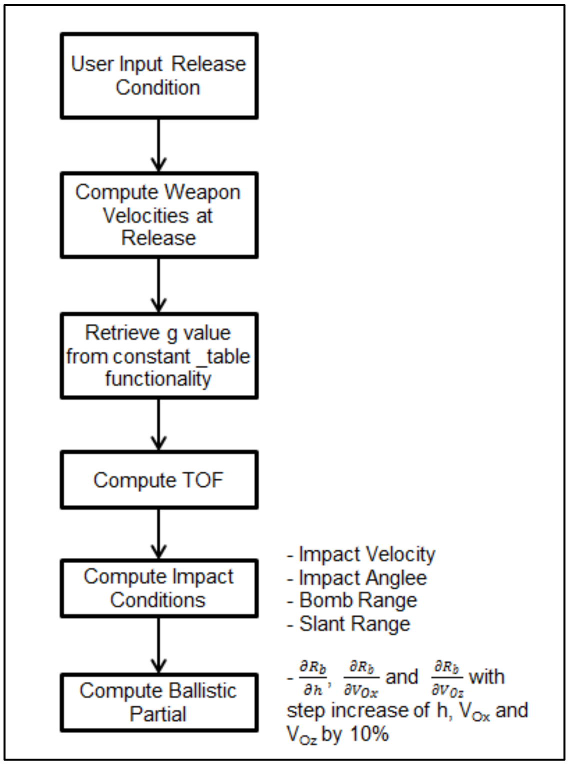


Figure 7. Flow diagram of programming of zero-drag trajectory module.

Note also the following:

1) Retrieving value for gravitational acceleration constant: The gravitational acceleration constant is not hard-coded in the zero-drag trajectory module. Instead, it is retrieved from the Excel database via a constant_table function in MATLAB. This provides a centralized place to view the constants.

2) Varying of inputs h, $\frac{\partial R}{\partial h}$, $\frac{\partial R_B}{\partial V_{Ox}}$ and $\frac{\partial R_B}{\partial V_{Oz}}$ in computation of the ballistic partials: The three inputs are each varied by 10% separately to compute the ballistic partials. This value of 10% can be changed subsequently if required in the coding.

2. Results of Zero-Drag Trajectory Program

The results of the zero-drag trajectory program were compared with a validated Excel program [2] and verified to be good, with the comparison summarized in Table 9.

Scenario	Output Variable	Validated Excel Program Results	Zero-Drag Trajectory Module
Scenario 1: Release Altitude: 8000ft Aircraft Speed: 400kts Dive Angle: 4° Ejection Velocity: 5 ft/s	TOF (s)	20.73	20.7323
	Impact Velocity (ft/s)	985.4	985.3857
	Impact Angle	46.91°	46.9145°
	Bomb Range (ft)	13955.45	13955
	Slant Range (ft)	16085.85	16084
	$\frac{\partial R_b}{\partial h}$	0.91317	0.9131
	$\frac{\partial R_b}{\partial V_{Ox}}$	20.7323	20.7323
	$\frac{\partial R_b}{\partial V_{Oz}}$	-19.3165	-19.3160
Scenario 2: Release Altitude: 3000ft Aircraft Speed: 450kts Dive Angle: 30° Ejection Velocity: 5 ft/s	TOF (s)	6.25	6.246
	Impact Velocity (ft/s)	877.53	877.5194
	Impact Angle	41.44°	41.4484°
	Bomb Range (ft)	4108.28	4108.3
	Slant Range (ft)	5087.04	5087

Table 9. Comparison of results for zero-drag trajectory module with validated Excel program.

C. HIGH-FIDELITY TRAJECTORY MODULE

The main difference between the zero-drag point mass module and the high-fidelity trajectory module is that the zero-drag module assumes that the weapon does not experience any drag as it falls towards the ground. Hence, we are able to form differential equations to compute the impact condition of the weapon. Conversely, the high-fidelity trajectory module assumes a non-linear drag effect on the weapon as it falls towards the ground. The impact condition of the weapon has to be computed by making multiple small time-step runs. At the end of each run, the acceleration, velocity and distance travelled by the weapon is computed for the next time-step run. The impact condition of the weapon can be calculated using the last known velocity and distance travelled by the weapon when the altitude of the weapon drops to zero. The equations to compute the acceleration, velocity and distance of the weapon at the end of each time-step are derived from the drag force (Drag) equation of $F_D = \frac{1}{2} S C_d \rho V^2$ and Newton's 2nd Law of $F = ma$.

In addition to the four inputs that are required for the zero-drag module (Table 7), the high-fidelity trajectory module requires additional inputs to account for the non-linear drag effect, as summarized in Table 10.

S/No	Input Parameter	Input Method	Description
1	Drag coefficient	Based on User Input of Weapon Type	The drag coefficient of the specific weapon chosen by user. It is assumed to be a constant regardless of Mach speed.
2	Weapon Diameter		The diameter of the specific weapon chosen by the user.
3	Weapon Mass		The mass of the specific weapon chosen by the user.
4	Air Density	Constant Table Functionality	The air-density is assumed to be constant regardless of altitude.

Table 10. Additional input for high-fidelity trajectory program.

The drag coefficient, weapon diameter, weapon mass and air density has to be determined based on the weapon type chosen by the operator prior to the onset of the

computation. The air density and drag coefficient are assumed to be constant with respect to altitude and weapon type, respectively, even though air density is affected by altitude and drag coefficient is affected by the MACH speed of the weapon. This assumption did not have a significant impact on the computed impact condition of the weapon.

Similar to the zero-drag trajectory module, the first step of the high-fidelity trajectory program is to compute the weapon velocities at release using the same equations.

$$V_{Ox} = V_a \cos\theta - V_e \sin\theta \tag{3.14}$$

$$V_{Oz} = V_a \sin\theta + V_e \cos\theta \tag{3.15}$$

$$V_O = \sqrt{{V_{Ox}}^2 + {V_{Oz}}^2} \tag{3.16}$$

Once the weapon velocities are computed, the acceleration of the weapon can be computed as follows. D and m are the diameter and mass of the weapon, respectively, while θ is the initial dive angle of the aircraft.

$$a_x = -\frac{1}{2}\frac{\pi D^2}{4}(C_d \rho {V_O}^2 \cos\theta)/m \tag{3.17}$$

$$a_z = g - \frac{1}{2}\frac{\pi D^2}{4}(C_d \rho {V_O}^2 \sin\theta)/m \tag{3.18}$$

Once the acceleration components are computed, the velocity, range and new height of the weapon at the end of a time step can be computed. The assumption here is that when the time-step used for the computation is small, the new velocity, range and height of the weapon can be computed assuming a linear relationship

$$V_x(t + dt) = V_x(t) + a_x * dt \tag{3.19}$$

$$V_z(t + dt) = V_z(t) + a_z * dt \tag{3.20}$$

$$V(t+dt) = \sqrt{V_x(t+dt)^2 + V_z(t+dt)^2} \tag{3.21}$$

where $V_x(t)$ and a_x are the existing velocity and acceleration prior to the start of the time step. $V_x(t+dt)$ is the velocity at the end of the time step.

$$x(t+dt) = x(t) + (V_x(t+dt) * dt) \tag{3.22}$$

$$h(t+dt) = h(t) - (V_z(t+dt) * dt) \tag{3.23}$$

$$\theta(t+dt) = tan^{-1}(\frac{v_y(t+dt)}{v_z(t+dt)}) \tag{3.24}$$

Here, the first cycle of the high fidelity module is completed. The cycle will be repeated with V(t+dt) replacing V(t), h(t+dt) replacing h(t) and θ(t+dt) replacing θ(t). The whole cycle is repeated until the weapon impacts the ground when h(t+dt)≤0.

1. High-Fidelity Trajectory Module in MATLAB

The high-fidelity trajectory module is modeled as a function. The input commands and the associated outputs for the module are shown in Table 11.

Input	`High_Fidelity_Trajectory_Model(release_altitude, dive_angle, ac_velocity, ejection_velocity, bomb_type)`
Output	`high_fidelity_trajectory_model_output = [Initial_Vertical_Velocity, Initial_Horizontal_Velocity, Cd, TOF, Horizontal_Impact_Velocity, Vertical_Impact_Velocity, Impact_Angle, Impact_Velocity, SR, Rb]`

Table 11. Input and output commands for high-fidelity trajectory module.

The high-fidelity trajectory module is modeled using the equations explained above, with a flowchart of the program shown in Figure 8. Note also:

1) Retrieving weapon parameters (weapon drag coefficient, mass and diameter): The parameters of the weapon are not hard coded in program

but are retrieved from the Excel data-base via an intermediate weapon parameter functionality. This provides users the ability to add or modify the data-base.

2) Retrieving air density constant: Like the gravitational acceleration constant, the air density constant is also retrieved from the Excel data-base via the constant_table function in MATLAB.

Step size for each cycle: There is a trade-off for the selection of the step size for each cycle. Choosing a very small step size will ensure higher accuracy as the equations are more likely to observe a linear relationship over the time-step. However, this will require a longer simulation time, especially when the weapon is released from a high altitude. For this module, the step size was hard coded as 0.001s, but it can be amended easily in the module as required.

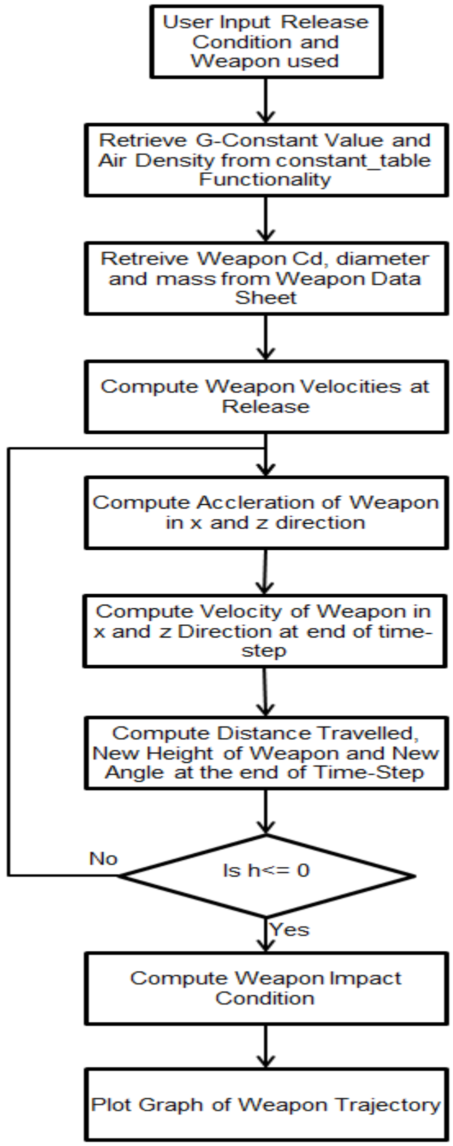


Figure 8. Flow diagram of high-fidelity trajectory module.

2. Results of High-Fidelity Trajectory Program

The results of the high-fidelity trajectory program were compared with an Excel program implemented using the exact same methodology and verified to be good. The high-fidelity trajectory program also produced the same result with the zero-drag trajectory program when the drag coefficient of the weapon was set as zero. The initial test condition is shown in Figure 9. The result for the high fidelity trajectory module is shown in Table 12.

Air-to-Surface System					
Program Type			Weapon Trajectory		
			Warhead Size		500lb (Low Drag)
[redacted]					
Aircraft Release Data					
Aircraft Speed	KTAS	450	Dive Angle	°	10
Release Altitude	ft	5000	Ejection Velocity	ft	5

Figure 9. Initial test conditions.

Trajectory Results				
Vertical Velocity	ft/s	136.8116		
Horizontal Velocity	ft/s	747.1031		
Drag Constant		0.11		
				Dragless
TOF	s	18.69	-	13.88313
Horizontal Velocity at Impact	ft/s	178.358	-	747.1031
Vertical Velocity at Impact	ft/s	372.008	-	583.4873
Impact Angle	deg	64.38477	-	37.98983
Impact Velocity	ft/s	412.5549	-	947.9559
Slant Range	ft	8523.041	-	11514.38
Ground Range	ft	6902.335	-	10372.13
$\delta R/\delta H$		1.25151	-	-
$\delta R/\delta Vx$		13.88313	-	-
$\delta R/\delta Vz$		-17.5203	-	-

Table 12. Results for high-fidelity trajectory module.

D. ADDITIONAL HIGH-FIDELITY TRAJECTORY MODULES

Two additional high-fidelity trajectory modules were required for the weapon effectiveness computation of stick delivery and cluster munitions. While the theory behind these two modules are the same as that for the high fidelity trajectory module, there are some differences in the modules as follows:

1) High-Fidelity Trajectory Module for Stick Delivery: For stick delivery, the high fidelity trajectory simulation has to be run twice to compute the

impact condition for the first and last weapon released. If the aircraft is flying with a dive angle, the impact conditions of the first and last weapon will not be the same. The interval between the weapon releases can be set by time or distance. The input commands and the associated outputs for the module are shown in Table 11.

Input	High_Fidelity_Trajectory_stick_delivery_Model(release_altitude, final_release_altitude, dive_angle, ac_velocity, ejection_velocity, Warhead_Size, nr, IV_Time)
Output	high_fidelity_trajectory_stick_delivery_model_output = [Cd, TOF_First_Wpn, Horizontal_Impact_Velocity_First_Wpn, Vertical_Impact_Velocity_First_Wpn, Impact_Angle_First_Wpn, Impact_Velocity_First_Wpn, SR_First_Wpn, Rb_First_Wpn, TOF_Last_Wpn, Horizontal_Impact_Velocity_Last_Wpn, Vertical_Impact_Velocity_Last_Wpn, Impact_Angle_Last_Wpn, Impact_Velocity_Last_Wpn, SR_Last_Wpn, Rb_Last_Wpn];

Table 13. Input and output commands of high-fidelity trajectory module for stick delivery.

2) <u>High-Fidelity Trajectory Module for Cluster Weapons:</u> For cluster weapons, the high fidelity trajectory simulation has to be modified. Initially, the trajectory will be that of the canister. However, when the cluster weapon is activated after a preset time or at a preset altitude, the remaining trajectory will be computed using the weapon parameters of the submunitions. The cluster weapon trajectory module also has to cater to the two scenarios of single drop or stick delivery of cluster weapons. The input command and the associated outputs for the module are shown in Table 14. A flowchart of the module is shown in Figure 25.

Input	`High_Fidelity_Trajectory_Cluster_Munitions_Model(release_altitude, final_release_altitude, dive_angle, ac_velocity, ejection_velocity, Warhead_Size,nr, IV_Time, Release_Type, functioning_time, functioning_altitude, Number_of_Weapons)`
Output1 (Single Weapon)	`high_fidelity_trajectory_cluster_munitions_model_output = [Impact_Angle_First_Wpn, Impact_Velocity_First_Wpn, SR_First_Wpn, Rb_First_Wpn];`
Output2 (Stick Delivery)	`high_fidelity_trajectory_cluster_munitions_model_output = [Average_Impact_Angle, Average_Impact_Velocity, Average_SR, Average_Rb, Rb_First_Wpn Rb_Last_Wpn];`

Table 14. Input and output commands of high-fidelity trajectory module for cluster weapons.

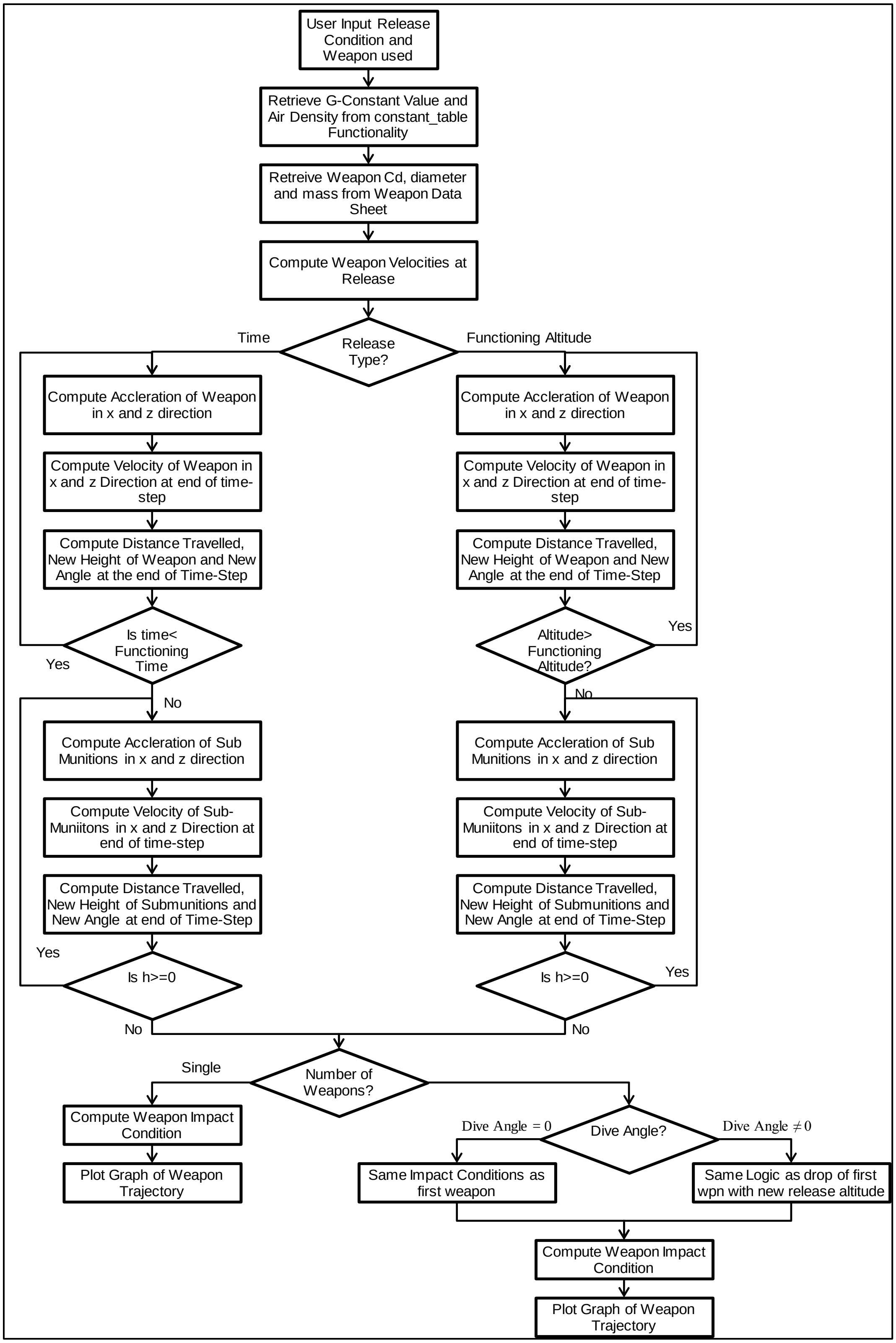


Figure 10. Flow diagram of trajectory module for cluster weapons.

IV. WEAPON DELIVERY ACCURACY

A. BACKGROUND

The weapon delivery accuracy module provides a quantitative measure of the ability of the weapon system to hit the target. The weapon accuracy is typically represented by the more common representation of CEP or the less commonly used REP and DEP. CEP, REP and DEP are defined as the radius, range and deflection boundaries, respectively, from the Desired Mean Point of Impact (DMPI) containing 50% of the impact point. CEP, REP and DEP serve as important inputs for computation of target damage probability for the various modules. The methodology to determine the delivery accuracy of the weapon is dependent on the following:

1) Weapon Type: The weapon type is in one of three different categories for the thesis: Laser Guided Weapons (LGB), GPS Guided Weapons and Unguided Weapons. While newer existing weapons such as the EGBU-12s can be equipped with all three modes, their accuracy will be dependent on the final mode being used prior to impact.

2) Mode of Weapon Release: The mode in which the weapon is being released will only affect unguided munitions. Guided weapons are not affected as their trajectory is guided by external sources such as the GPS signal from GPS satellites or the reflected laser signals from the target. The three weapon release modes for unguided weapons being modeled in this program are CCRP, CCIP and BOC.

The errors associated with each weapon type and weapon release mode are shown in 0Each of the different methodologies will be described. The materials in this chapter are taken from Chapter 4 and Chapter 5 of [1].Additional theoretical information can be found in the same source.

Weapon Type	Weapon Release Mode	Variables affecting CEP/REP/DEP																
		Aircraft Sensor Variable Error										Targeting Pod Sensor Variable Error		Wpn Guidance Error	GPS Error	Target Location Error		
		V_x (ft/s)	V_y (ft/s)	V_z (ft/s)	ω, mrad (roll)	λ, mrad (pitch)	Ψ, mrad (yaw)	H, ft	X_a, ft	Y_a, ft	Z_a, ft	θ, mrad	SR, ft			X_t, ft	Y_t, ft	Z_t, ft
Guided LGB	N.A.	-	-	-	-	-	-	-	-	-	-	-	-	Y	-	-	-	-
Guided GPS	N.A.	-	-	-	-	-	-	-	-	-	-	-	-	Y	Y	Y	Y	Y
Unguided	CCRP	Y	Y	Y	Y	Y	Y	-	-	-	-	Y	Y	-	-	-	-	-
	CCIP	Y	Y	Y	Y	Y	Y	Y	-	-	-	-	-	-	-	-	-	-
	BOC	Y	Y	Y	Y	Y	Y		Y	Y	Y	-	-	-	-	Y	Y	Y

Table 15. Variables affecting CEP/REP/DEP of weapon.

B. UNGUIDED CCRP

CCRP is a mode where the Fire Control Computer (FCC) continuously computes the release point after the pilot designates the target. The accuracy of CCRP mode is primarily affected by errors of onboard sensors estimating the aircraft state and the targeting pod sensor error variable. Each of these error variables can be translated into an error sigma in the range and deflection direction. The total error sigma in range and deflection is used to compute the REP and DEP of the weapon, respectively. The CEP can then be computed based on the REP and DEP of the weapon. The computation methodology is as follows.

Error in Range Direction due to Targeting Pod Sensor

$$\text{Error due to LOS angle, } \sigma_{X-\theta} = -[h_D + \left(R_D * \frac{\partial R_B}{\partial h_R}\right)]\sigma_\theta \tag{4.1}$$

$$\text{Error due to SR, } \sigma_{X-SR} = \frac{1}{SR_D}[R_D - \left(h_D * \frac{\partial R_B}{\partial h_R}\right)]\sigma_{SR} \tag{4.2}$$

Error in Range Direction due to Aircraft Sensor

$$\text{Error due to velocity} - x,\ \sigma_{X-VX} = [\frac{\partial R_B}{\partial V_x} + T_{DR}]\sigma_{VX} \tag{4.3}$$

$$\text{Error due to velocity} - z, \sigma_{X-VZ} = [T_{DR}\frac{\partial R_B}{\partial h_R} + \frac{\partial R_B}{\partial V_z}]\sigma_{VZ} \tag{4.4}$$

$$\text{Error due to pitch angle, } \sigma_{X-\lambda} = -[h_D + R_D\frac{\partial R_B}{\partial h_R}]\sigma_\lambda \tag{4.5}$$

Total Error in Range Direction

$$\sigma_X = \sqrt{{\sigma_{X-\theta}}^2 + {\sigma_{X-SR}}^2 + {\sigma_{X-VX}}^2 + {\sigma_{X-VZ}}^2 + {\sigma_{X-\lambda}}^2} \tag{4.6}$$

$$REP = 0.6745 * \sigma_X \tag{4.7}$$

Error in Deflection Direction due to Aircraft Sensor

$$\text{Error due to velocity} - y,\ \sigma_{Y-VY} = TOF * \sigma_{VY} \tag{4.8}$$

$$\text{Error due to roll angle, } \sigma_{Y-\omega} = v_{ejec} * TOF * \sigma_\omega \tag{4.9}$$

$$\text{Error due to yaw angle, } \sigma_{Y-\Psi} = R_D * \sigma_\Psi \tag{4.10}$$

Total Error in Deflection Direction

$$\sigma_Y = \sqrt{\sigma_{Y-VY}^2 + \sigma_{Y-\omega}^2 + \sigma_{Y-\Psi}^2} \tag{4.11}$$

$$\text{DEP} = 0.6745 * \sigma_Y \tag{4.12}$$

CEP Computation

$$\text{CEP} = 0.582 \max(\sigma_X, \sigma_Y) + 0.617 \min(\sigma_X, \sigma_Y) \tag{4.13}$$

1. CCRP Module in MATLAB

The CCRP module is modeled as a function using the equations explained above. The input commands and the associated outputs for the CCRP Module are depicted in Table 16. A flowchart of the module is shown in Figure 11.

Input	`CCRP_Model( release_altitude, dive_angle, ac_velocity, ejection_velocity, Warhead_Size, RD, Targeting_Pod, Air_Platform_Type, Sigma_b )`
Output	`CCRP_output = [REP, DEP, CEP, x_bd, x_br, Adjusted_REP, Adjusted_DEP];`

Table 16. Input and output commands for CCRP module.

Note also the dive angle for CCRP mode is set as 0°. Any other dive angle would heavily complicate how the time to release of the weapon is computed:

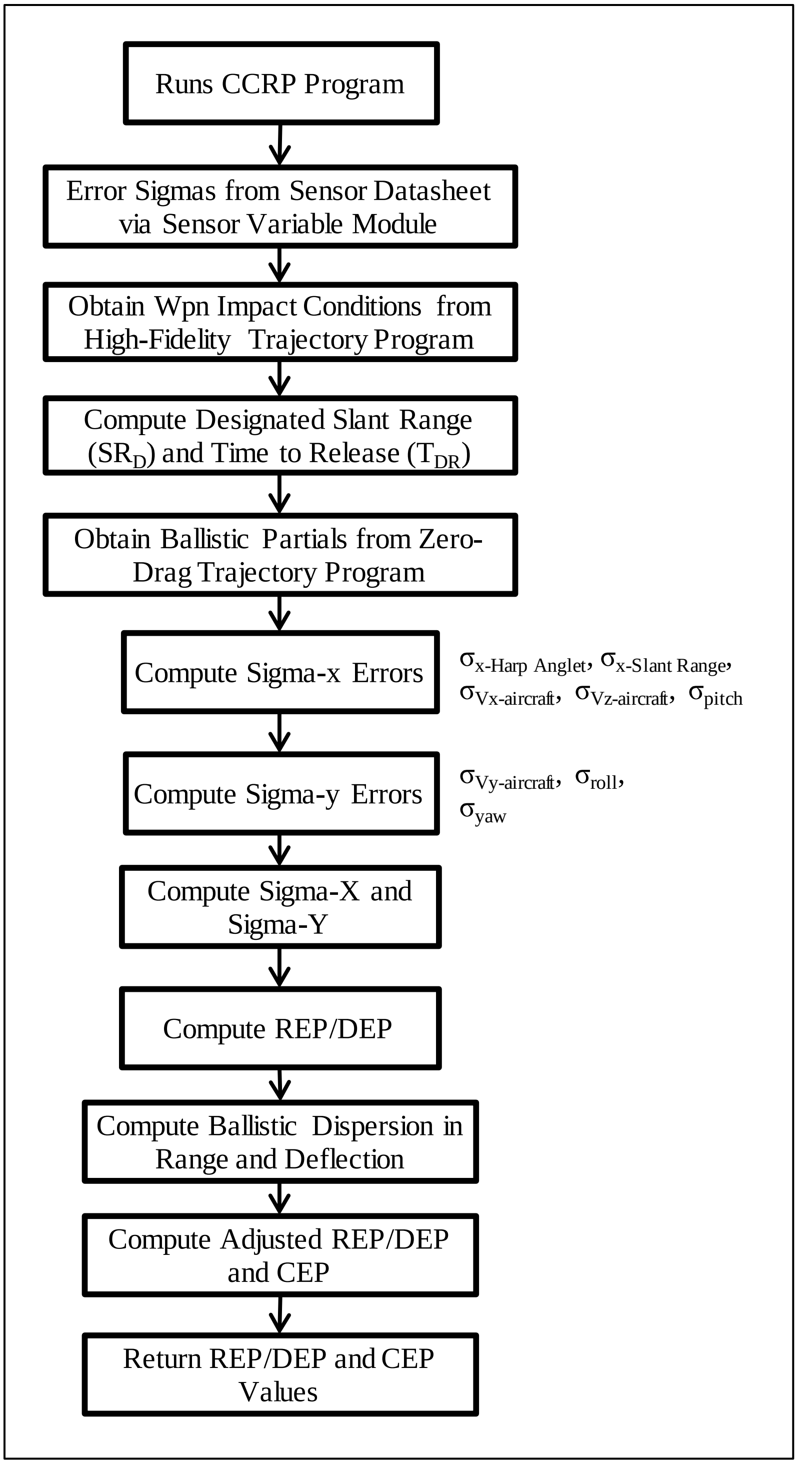


Figure 11. Flow diagram of CCRP module.

2. Results of CCRP Program

The results of the CCRP program were compared with the examples in [1] after changing the impact angle conditions and were verified to be good.

C. UNGUIDED CCIP

CCIP is a mode in which the FCC continuously computes and projects onto the Heads-Up-Display (HUD) the impact point of the weapon should the pilot release the weapon at that instant. The accuracy of CCIP mode is primarily affected by the errors of onboard sensors estimating the aircraft state such as the altitude of the aircraft. Each of these error variables can be translated into an error sigma in the range and deflection direction. The total error sigma in range and deflection is used to compute the REP and DEP of the weapon, respectively. The CEP can then be computed based on the REP and DEP of the weapon. The computation methodology is as follows.

Error in Range Direction due to Aircraft Sensor

$$\text{Error due to altitude, } \sigma_{X-h} = -\left[\frac{R_B}{h} - \frac{\partial R_B}{\partial h}\right]\sigma_h \tag{4.14}$$

$$\text{Error due to velocity} - x,\ \sigma_{X-VX} = \frac{\partial R_B}{\partial V_x}\sigma_{VX} \tag{4.15}$$

$$\text{Error due to velocity} - z,\ \sigma_{X-VZ} = \frac{\partial R_B}{\partial V_z}\sigma_{VZ} \tag{4.16}$$

$$\text{Error due to pitch angle, } \sigma_{X-\lambda} = -\frac{SR^2}{h}\sigma_\lambda \tag{4.17}$$

Total Error in Range Direction

$$\sigma_X = \sqrt{{\sigma_{X-h}}^2 + {\sigma_{X-VX}}^2 + {\sigma_{X-VZ}}^2 + {\sigma_{X-\lambda}}^2} \tag{4.18}$$

$$\text{REP} = 0.6745 * \sigma_X \tag{4.19}$$

Error in Deflection Direction due to Aircraft Sensor

$$\text{Error due to velocity} - y,\ \sigma_{Y-VY} = \text{TOF} * \sigma_{VY} \tag{4.20}$$

$$\text{Error due to roll angle, } \sigma_{Y-\omega} = v_{ejec} * \text{TOF} * \sigma_\omega \tag{4.21}$$

$$\text{Error due to yaw angle, } \sigma_{Y-\Psi} = R_B * \sigma_\Psi \tag{4.22}$$

Total Error in Deflection Direction

$$\sigma_Y = \sqrt{\sigma_{Y-VY}{}^2 + \sigma_{Y-\omega}{}^2 + \sigma_{Y-\Psi}{}^2} \tag{4.23}$$

$$\mathrm{DEP} = 0.6745 * \sigma_Y \tag{4.24}$$

CEP Computation

$$\mathrm{CEP} = 0.582 \max(\sigma_X, \sigma_Y) + 0.617 \min(\sigma_X, \sigma_Y) \tag{4.25}$$

1. CCIP Module in MATLAB

The CCIP module is modeled as a function using the equations explained above. The input commands and the associated outputs for the CCIP module are depicted in Table 17. A flowchart of the module is shown in Figure 12.

Input	`CCIP_Model( release_altitude, dive_angle, ac_velocity, ejection_velocity, Warhead_Size, Air_Platform_Type, Sigma_b)`
Output	`CCIP_output = [REP, DEP, CEP, x_bd, x_br, Adjusted_REP, Adjusted_DEP];`

Table 17. Input and output commands for CCIP module.

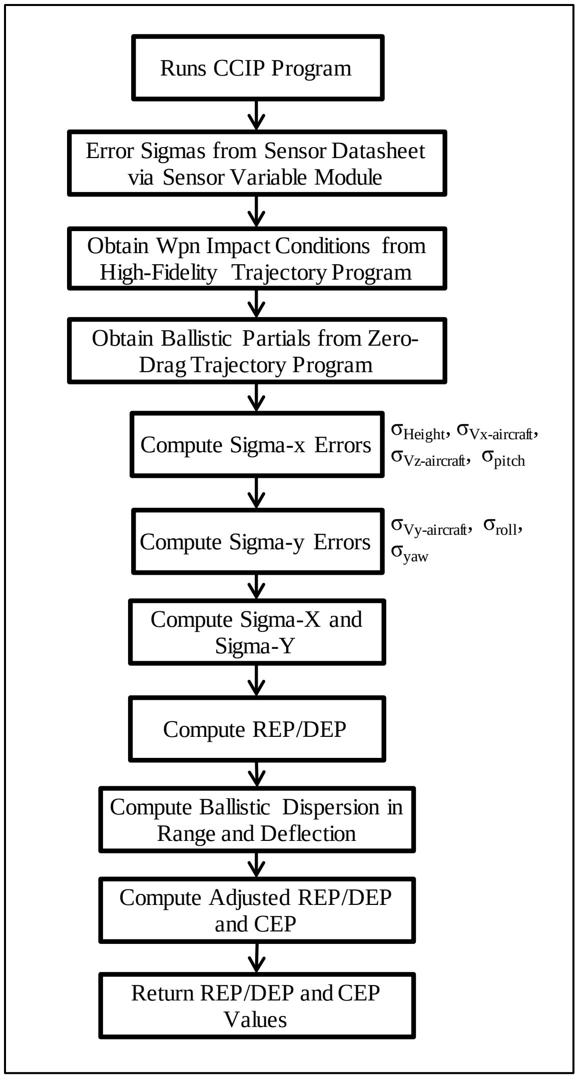


Figure 12. Flow diagram of CCIP module.

2. Results of CCIP Program

The results of the CCIP program were compared with the examples in [1] after changing the impact conditions in the program and were verified to be good.

D. UNGUIDED BOC

BOC is a mode in which the weapon is released by the FCC based on the perceived location of the target. The accuracy of BOC mode is primarily affected by the aircraft error variable and the target location error variable. Each of these error variables can be translated into an error sigma in the range and deflection direction. The total error

sigma in range and deflection is used to compute the REP and DEP of the weapon, respectively. The CEP can then be computed based on the REP and DEP of the weapon. The computation methodology is as follows.

Error in Range Direction due to Aircraft Sensor

$$\text{Error due to aircraft x location, } \sigma_{X-a} = \sigma_{X-aircraft} \tag{4.26}$$

$$\text{Error due to aircraft z location, } \sigma_{Z-a} = \sigma_{Z-aircraft} * \frac{\partial R_B}{\partial h} \tag{4.27}$$

$$\text{Error due to velocity} - \text{x, } \sigma_{X-VX} = \frac{\partial R_B}{\partial V_x} \sigma_{VX} \tag{4.28}$$

$$\text{Error due to velocity} - \text{z, } \sigma_{X-VZ} = \frac{\partial R_B}{\partial V_z} \sigma_{VZ} \tag{4.29}$$

$$\text{Error due to pitch angle, } \sigma_{X-\lambda} = -\frac{SR^2}{h} \sigma_{\lambda} \tag{4.30}$$

Error in Range Direction due to Target Location Error

$$\text{Error due to target x location, } \sigma_{X-t} = \sigma_{X-target} \tag{4.31}$$

$$\text{Error due to target z location, } \sigma_{Z-t} = \sigma_{Z-target} * \frac{\partial R_B}{\partial h} \tag{4.32}$$

Total Error in Range Direction

$$\sigma_X = \sqrt{{\sigma_{X-a}}^2 + {\sigma_{Z-a}}^2 + {\sigma_{X-t}}^2 + {\sigma_{Z-t}}^2 + {\sigma_{X-VX}}^2 + {\sigma_{X-VZ}}^2 + {\sigma_{X-\lambda}}^2} \tag{4.33}$$

$$REP = 0.6745 * \sigma_X \tag{4.34}$$

Error in Deflection Direction due to Aircraft Sensor

$$\text{Error due to aircraft y location, } \sigma_{Y-a} = \sigma_{Y-aircraft} \tag{4.35}$$

$$\text{Error due to velocity} - \text{y, } \sigma_{Y-VY} = TOF * \sigma_{VY} \tag{4.36}$$

$$\text{Error due to roll angle, } \sigma_{Y-\omega} = v_{ejec} * TOF * \sigma_{\omega} \tag{4.37}$$

$$\text{Error due to yaw angle, } \sigma_{Y-\Psi} = R_B * \sigma_{\Psi} \tag{4.38}$$

Error in Deflection Direction due to Target Location Error

$$\text{Error due to target y location, } \sigma_{Y-t} = \sigma_{Y-target} \tag{4.39}$$

Total Error in Deflection Direction

$$\sigma_Y = \sqrt{{\sigma_{Y-a}}^2 + {\sigma_{Y-t}}^2 + {\sigma_{Y-VY}}^2 + {\sigma_{Y-\omega}}^2 + {\sigma_{Y-\Psi}}^2} \qquad (4.40)$$

$$\mathrm{DEP} = 0.6745 * \sigma_Y \qquad (4.41)$$

CEP Computation

$$\mathrm{CEP} = 0.582 \max(\sigma_X, \sigma_Y) + 0.617 \min(\sigma_X, \sigma_Y) \qquad (4.42)$$

1. BOC Module in MATLAB

The BOC module is modeled as a function using the equations explained above. The input commands and the associated outputs for the BOC module are depicted in Table 18. A flowchart of the module is shown in Figure 11.

Input	`BOC_Model( release_altitude, dive_angle, ac_velocity, ejection_velocity, Warhead_Size, Air_Platform_Type, Sigma_b, Target_Position_Error, Target_Position_Error_x, Target_Position_Error_y, Target_Position_Error_z)`
Output	`BOC_output = [REP, DEP, CEP, x_bd, x_br, Adjusted_REP, Adjusted_DEP];`

Table 18. Input and output commands for BOC module.

Note also the TLE input to the BOC module can be manually input by user or retrieved from the Excel default values.

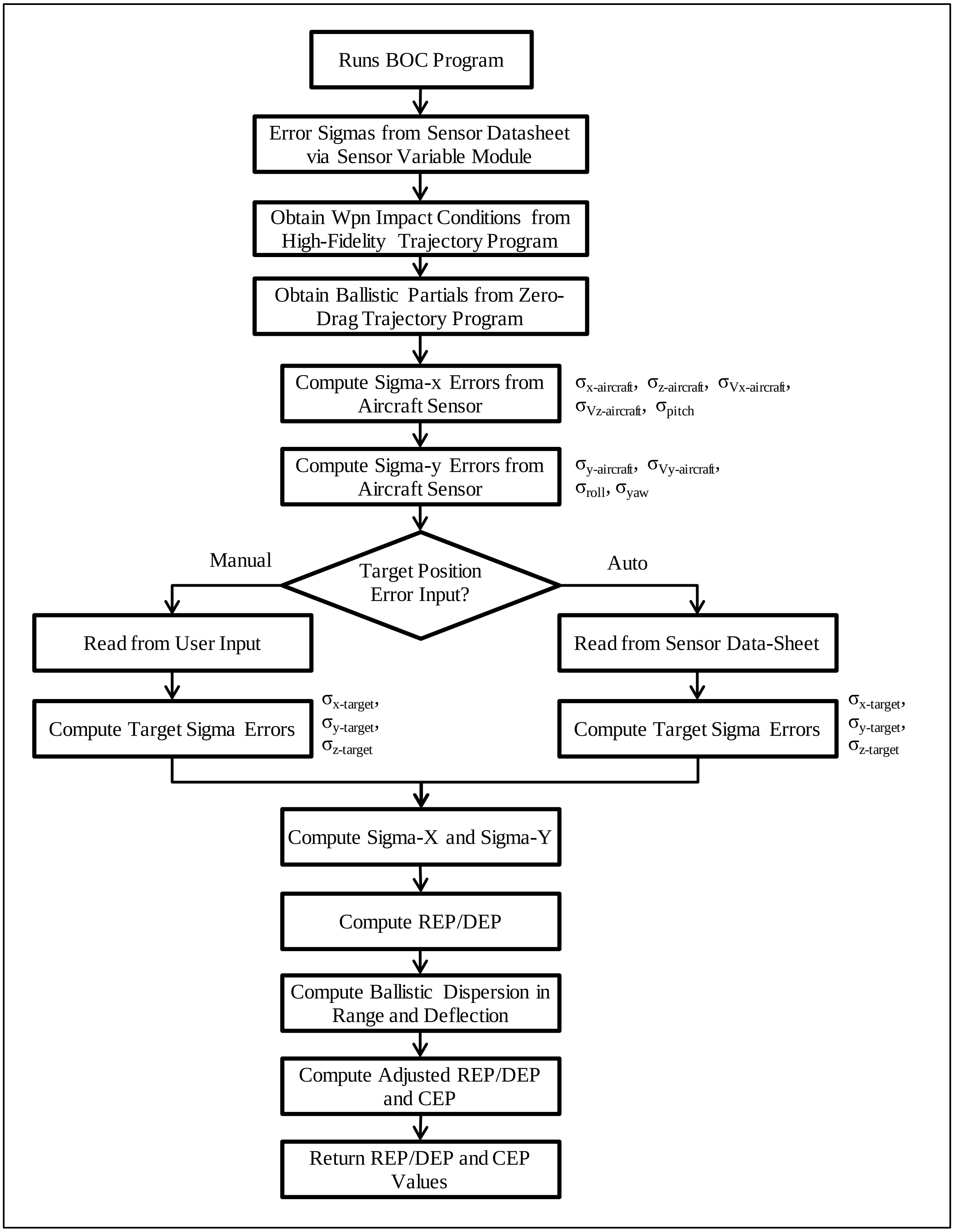


Figure 13. Flow diagram of BOC module.

2. Results of BOC Program

The results of the BOC program were compared with the examples in [1] after amending the trajectory inputs in the program and were verified to be good.

E. LASER GUIDED WEAPONS

There is no weapon accuracy program required for laser guided weapons due to the simplicity of computing the DEP and REP of such weapons. As mentioned previously, weapons are not dependent on the effects of the errors from onboard aircraft and targeting pod. Instead, the accuracy of the laser guided weapons is typically affected by the accuracy of its guidance system. This accuracy is typically denoted as the CEP of the weapon by weapons manufacturer. Hence, with the provided CEP specification, the DEP and REP of the weapon can be computed as follows:

$$DEP = REP = \frac{CEP}{1.7456} = 0.573 * CEP \tag{4.43}$$

F. GPS GUIDED WEAPONS

The accuracy of GPS guided weapons is mainly determined by the following:

1) Accuracy of the GPS of the weapon.

2) Accuracy of the GPS coordinates of the target, otherwise known as Target Location Error.

3) Ability of guidance and control system (GC) to maneuverer the GPS weapon to the target, which is denoted by the CEP of the weapon.

The computation methodology for accuracy of GPS weapons assumes in general that the errors in range and deflection are the same. Hence, the error in range and deflection, denoted by Δx and Δy, can be computed as follows for each of the three factors affecting the accuracy of GPS weapons.

<u>Error due to the GPS of the Weapon</u>

$$\Delta \mathrm{x}_{\mathrm{GPS}} = \Delta \mathrm{y}_{\mathrm{GPS}} = \Delta \mathrm{GPS}_{\mathrm{H}} \text{ where } \Delta \mathrm{GPS}_{\mathrm{H}} \text{ is the horizontal GPS error} \tag{4.44}$$

$$\Delta z_{GPS} = \Delta GPS_v \; where \; \Delta GPS_H \; is \; the \; vertical \; GPS \; error \tag{4.45}$$

Error due Target Location Error (Similar to BOC Computation)

$$\Delta x_{TLE} = \Delta TLE_x \tag{4.46}$$

$$\Delta y_{TLE} = \Delta TLE_y \tag{4.47}$$

$$\Delta z_{TLE} = \frac{\partial R_B}{\partial h} * \Delta TLE_z \tag{4.48}$$

Error due GC of the Weapon

$$\Delta x_{GC} = \Delta y_{GC} = \Delta GC \tag{4.49}$$

Computation of REP and DEP

$$REP = \sqrt{\Delta GC^2 + \Delta x_{TLE}{}^2 + \Delta x_{GPS}{}^2 + \sqrt{\frac{\Delta z_{TLE}{}^2 + \Delta z_{GPS}{}^2}{\tan(\text{Impact Angle})}}} \tag{4.50}$$

$$DEP = \sqrt{\Delta GC^2 + \Delta y_{TLE}{}^2 + \Delta y_{GPS}{}^2} \tag{4.51}$$

While the equation above provides a more accurate REP computation, a decision was made to omit the vertical component of the computation to simplify what users have to enter when selecting GPS weapons. Hence, the REP computation is simplified as follows:

$$REP = \sqrt{\Delta GC^2 + \Delta x_{TLE}{}^2 + \Delta x_{GPS}{}^2} \tag{4.52}$$

G. MANUAL INPUT OF CEP

When the user is performing target damage computation for bombs, the program allows the users to manually input the CEP of the weapon if the information is known. There is a need to translate the CEP entered into REP and DEP, because these are the actual inputs used by the weapons effectiveness computation modules. This is done so using the equation below:

$$DEP = REP = \frac{CEP}{1.7456} = 0.573 * CEP \tag{4.53}$$

H. TREATMENT OF BALLISTIC DISPERSION ERROR

Typically each batch of munitions has a ballistic dispersion error that is specified by the weapon's manufacturer. For individually released weapons, these ballistic dispersion errors should be included to in the weapon accuracy computation to provide the equivalent REP and DEP. In the program, the ballistic dispersion error is a user input and affects the REP and DEP using the equations below.

$$x_{bd} = \frac{0.6745 * \mathrm{SR} * \sigma_b}{1000} \tag{4.54}$$

$$\mathrm{x_{br}} = \frac{0.6745 * \mathrm{SR} * \sigma_b}{1000 * \sin\theta_i} \tag{4.55}$$

$$\mathrm{REP}' = \sqrt{\mathrm{REP}^2 + \mathrm{x_{br}}^2} \tag{4.56}$$

$$\mathrm{DEP}' = \sqrt{\mathrm{DEP}^2 + \mathrm{x_{bd}}^2} \tag{4.57}$$

The results of the equivalent REP and REP computation were compared with the examples in [1] and verified to be good.

V. SINGLE WEAPON UNITARY/AREA TARGET PROGRAM

A. SINGLE WEAPON UNITARY TARGET PROGRAM

1. Background

The single weapon unitary target program computes the probability of damage of employing a single AS weapon against a unitary target. This probability of damage is being represented as the Single Sortie Probability of Damage (SSPD). The weapon of interest can be an unguided weapon, laser guided weapon or GPS guided weapon. The materials in this chapter are taken from Chapter 11 of [1]. Additional theoretical information can be found in the same source.

2. Weapon Effectiveness of Unguided Weapons

There are two types of effects for unguided weapons. The first is from a blast warhead where the effects are not affected by the impact angle as blast propagates radially from the point of detonation regardless of impact angle. The second is from a fragmentation warhead where the effects are affected by impact angle.

The weapon effectiveness of the weapon can be computed as a function of the accuracy function g(x) and the damage function c(x) in the form of

$$\mathrm{E}[\mathrm{c}(\mathrm{x})] = \int_{-\infty}^{\infty} \mathrm{c}(\mathrm{x})\mathrm{g}(\mathrm{x})\mathrm{dx} \tag{5.1}$$

The accuracy function is a function of the REP/DEP of the weapon, while the damage function is a function of weapon’s effectiveness index, which is represented as the Mean Area of Effectiveness (MAE) for this program.

Weapons Effectiveness Computation for Fragmentation Kill Mode

The damage function of a fragmentation warhead is represented by the Carleton Damage function and defined by the effective target length LET’ and effective target width WET’.

$$\text{Impact Angle Parameter, } \mathrm{a} = \max(1 - 0.8\cos(\text{Impact Angle}), 0.3) \tag{5.2}$$

$$\mathrm{LET}' = 1.128 * \sqrt{\mathrm{MAE} * \mathrm{a}} \tag{5.3}$$

$$\mathrm{WET}' = \frac{\mathrm{LET}'}{a} \tag{5.4}$$

The accuracy of the weapon is defined by the REP and DEP of the weapon. Hence, the SSPD of the weapon in the x and y axis can be computed as follows:

$$\mathrm{SSPD_x} = \frac{\mathrm{LET}}{\sqrt{17.6*\mathrm{REP}^2+\mathrm{LET}^2}} \tag{5.5}$$

$$\mathrm{SSPD_y} = \frac{\mathrm{WET}}{\sqrt{17.6*\mathrm{DEP}^2+\mathrm{WET}^2}} \tag{5.6}$$

<u>Weapons Effectiveness Computation for Blast Kill Mode</u>

The damage function of a blast warhead is represented by a square on the ground plane and defined by LET and WET.

$$\mathrm{LET} = \mathrm{WET} = \sqrt{\mathrm{MAE}} \tag{5.7}$$

The accuracy of the weapon is defined by σ_x and σ_y, which can be computed from the REP and DEP of the weapon.

$$\sigma_x = \mathrm{REP}/0.6745 \tag{5.8}$$

$$\sigma_y = \mathrm{DEP}/0.6745 \tag{5.9}$$

The SSPD of the weapon in the x and y axis can then be computed as follows:

$$\mathrm{SSPD_x} = \mathrm{normcdf}\left(\frac{\mathrm{LET}}{2}, 0, \sigma_x\right) - \mathrm{normcdf}(-\frac{\mathrm{LET}}{2}, 0, \sigma_x) \tag{5.10}$$

$$\mathrm{SSPD_y} = \mathrm{normcdf}\left(\frac{\mathrm{LET}}{2}, 0, \sigma_y\right) - \mathrm{normcdf}(-\frac{\mathrm{LET}}{2}, 0, \sigma_y) \tag{5.11}$$

<u>Computation of SSPD</u>

The SSPD of both blast and fragmentation warhead can then be computed in the same way as follows:

$$\mathrm{SSPD} = \mathrm{SSPD_x} * \mathrm{SSPD_y} * \text{Weapon Reliability} \tag{5.12}$$

3. Weapon Effectiveness of Guided Weapons

The methodology to compute the weapon effectiveness for guided weapons is similar with unguided weapons with the following differences. Unlike unguided

weapons, guided weapons do not follow a strict Gaussian distribution. Rather, they have a higher number of hits and higher number of gross errors (outliers). Hence, the SSPD computation is comprised of two parts, as represented by the equation below. The first part denoted represents the Gaussian distribution behavior of the weapon and is denoted by SSPD1. The second part accounts for the higher number of hits using a guided weapon and is denoted as SSPD2.

$$\mathrm{SSPD} = \mathrm{SSPD1} * \mathrm{P_{NM}} + \mathrm{SSPD2} * \mathrm{P_{HIT}} \tag{5.13}$$

P_{NM} and P_{HIT} are the weighing factors used to balance the direct hits and Gaussian miss distribution unique for guided weapons. The addition of P_{NM} and P_{HIT} can be equal to or smaller than 1 due to the presence of the outliers gross errors. The computation of SSPD1 is the same as the computation of SSPD for unguided weapons. The SSPD2 can be computed by using the same formula with the exception that the miss distance deviation is set as zero for blast effect weapons, and the REP/DEP is set as zero for fragmentation effect weapons to represent a zero miss distance for the P_{HIT} component.

<u>SSPD2 Computation for Blast Effect</u>

$$\mathrm{SSPD2_x} = \mathrm{normcdf}\left(\frac{\mathrm{LET}}{2}, 0{,}0\right) - \mathrm{normcdf}(-\frac{\mathrm{LET}}{2}, 0{,}0) \tag{5.14}$$

$$\mathrm{SSPD2_y} = \mathrm{normcdf}\left(\frac{\mathrm{WET}}{2}, 0{,}0\right) - \mathrm{normcdf}(-\frac{\mathrm{WET}}{2}, 0{,}0) \tag{5.15}$$

Note that for practical reasons, we have to change the miss distance deviation to at least 0.001 as a 0 value would result in an erroneous result in MATLAB when using the normcdf command.

<u>SSPD2 Computation for Fragmentation Effect</u>

$$\mathrm{SSPD_x} = \frac{\mathrm{LET}}{\sqrt{17.6 * \mathrm{REP}^2 + \mathrm{LET}^2}} \tag{5.16}$$

$$\mathrm{SSPD_y} = \frac{\mathrm{WET}}{\sqrt{17.6 * \mathrm{DEP}^2 + \mathrm{WET}^2}} \tag{5.17}$$

As REP and DEP are set as zero, the $SSPD_x$ and $SSPD_x$ will always result in a value of 1.

4. Single Weapon Unitary Target Module in MATLAB

The single weapon unitary target module is modeled as a function using the equations explained in the previous section. The input commands and the associated outputs for the module are depicted in Table 19. A flowchart of the module is shown in Figure 12.

Input	`Single_Wpn_Unitary_Target_Model(MAE, Kill_Mode, REP, DEP, Impact_Angle, Weapon_Reliability, Weapon_Type, P_Near_Miss, P_Hit)`
Output for Unguided Weapons	`Single_Wpn_Unitary_Target_output = [LET, WET, SSPD_x, SSPD_y,SSPD];`
Output for Guided Weapons	`Single_Wpn_Unitary_Target_output = [LET, WET, SSPD1_x, SSPD1_y,SSPD1, SSPD2_x, SSPD2_y,SSPD2,SSPD];`

Table 19. Input and output commands for single weapon unitary target module.

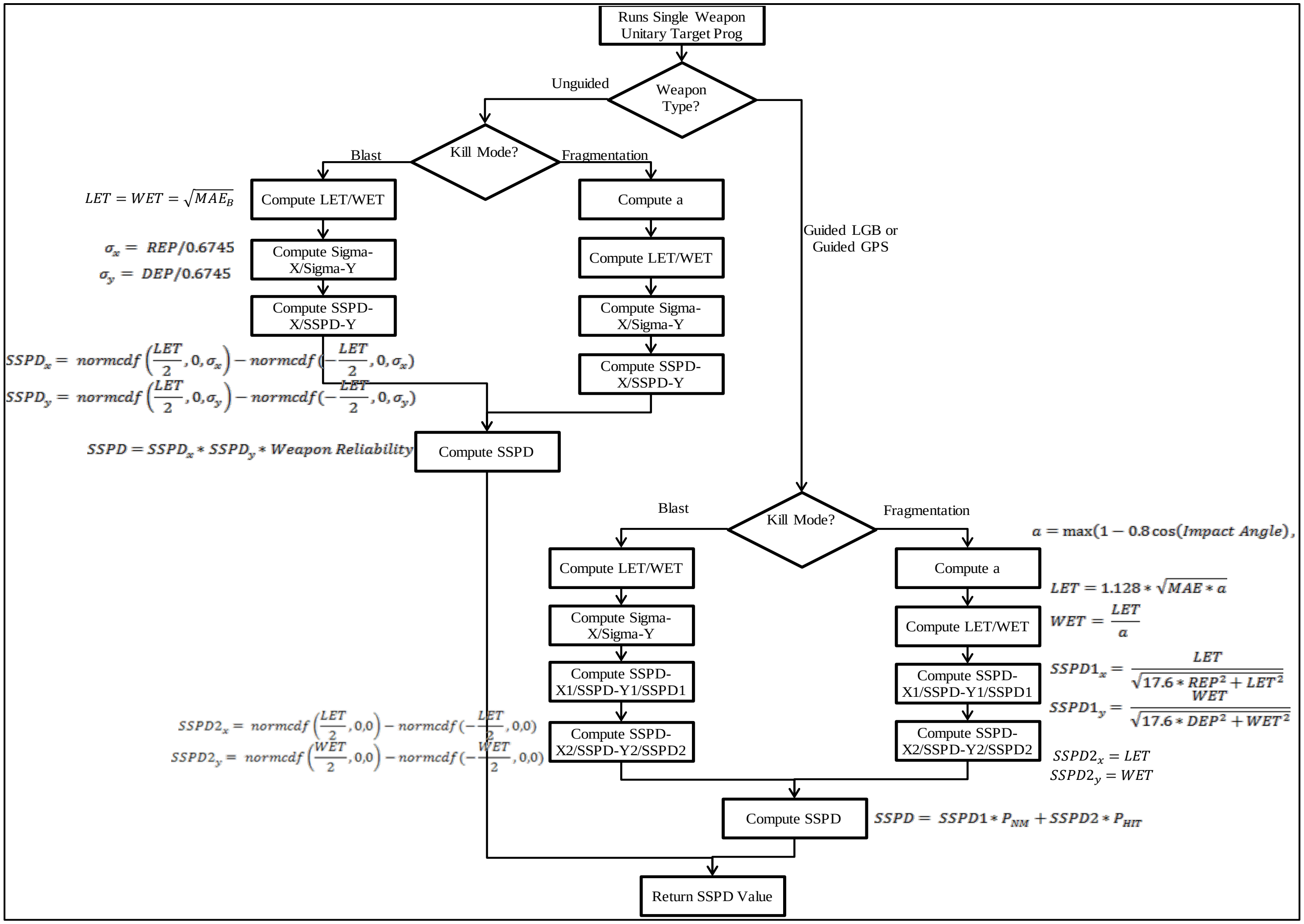


Figure 14. Flow diagram of single weapon unitary target module.

5. Results of Single Weapon Unitary Target Program

The results of the single weapon unitary target program were compared with an Excel program provided by Prof. Driels [3] and were verified to be good. A print-screen of the result provided by Prof. Driels is illustrated in Figure 15. A print-screen of the result from the single weapon unitary target program is illustrated in Figure 16.

15	Drag-L=low, H=high		L				
16	Ballistic dispersion, mils	Sigma-b	5.00				
17	Reliability	R	0.94				
18							
19	**AIRCRAFT RELEASE DATA**						
20							
21	Airspeed, knots	V	450.00		760	ft/sec	
22	Dive angle, degrees	theta	30.00		0.524	rad	
23	Release altitude, ft	yi	3000				
41			Ground ft	Normal ft	mils	REP/DEP	P
42	mils,spec	CEP	5.00	0	0		
43	Deflection error probable, ft	DEP	2.86	0	0	0.000	2.864
44	Range error probable, ft	REP	2.86	0	0	0.000	2.864
45	Ballistic error, normal plane,	sigma_b	5.00				
46	Ballistic error, deflection, ft	sigma-	14.70				
47	Ballistic error range, ft	sigma-	16.53				
48	Adjusted DEP, ft	DEP'	14.97				
49	Adjusted REP, ft	REP'	16.78				
50							
51	**WEAPON EFFECTIVENESS**						
52							
53	Effectiveness index type			MAE_b	MAE_f		
54	Effectiveness value	EI		0	10000		
55	Length/width ratio (frag only)	a	0.634				WET/LET
56	Effective target width ft	WET		0.000	141.700		141.700
57	Effective target length ft	LET		0.000	89.794		89.794
58	Effective target area ft^2	AET		0.000	10000		10000
59							
60	**PD_1 CALCULATION**						
61		RANGE		DEFLECTION			
62	Sigma (ft)	24.872		22.197			
63	PD1	0.787		0.914			
64	**Combined PD1**	**0.676**					

Figure 15. Results from Prof. Driels' Excel program. From [3]

Simulation Status		Completed	2	
Trajectory Results				
Vertical Velocity	ft/s	379.755		
Horizontal Velocity	ft/s	657.755		
Drag Constant		0.25		
				Dragless
TOF	s	10.594	-	6.246785
Horizontal Velocity at Impact	ft/s	140.9552	-	657.755
Vertical Velocity at Impact	ft/s	273.6742	-	580.7391
Impact Angle	deg	62.74939	-	41.44164
Impact Velocity	ft/s	307.8408	-	877.4391
Slant Range	ft	4360.688	-	5087.502
Ground Range	ft	3164.743	-	4108.854
δR/δH		1.116857	-	-
δR/δVx		6.246785	-	-
δR/δVz		-6.70851	-	-
		-	-	-
		-	-	-
		-	-	-
		-	-	-

Target Damage Computation Type	1	Single Weapon Unitary Targe		Error Code
Guidance Type	3	Unguided		Results
Weapon Effectiveness				
Effectiveness Index Type		MAE_b	MAE_f	
Effectiveness Value	**EI**		10000	
Length/Width Ratio (Frag Only)	**a**		0.633693279	
Effective Target Width, ft	**WET**		141.6999017	
Effective Target Length, ft	**LET**		89.79427539	
Effective Target Area, ft^2	**AET**		12723.84	
SSPD Calculation		**Range**	**Deflection**	
Sigma (ft)				
SSPD		0.786831497	0.914102227	
Combined SSPD		0.676089758		

Figure 16. Results from single weapon unitary target program.

B. SINGLE WEAPON AREA TARGET PROGRAM

1. Background

The single weapon area target program computes the probability of damage when employing a single AS weapon against an area target. This probability of damage is represented as the Expected Fractional Damage (EFD). The weapon of interest can be an unguided weapon, laser guided weapon (LGB) or GPS guided weapon. The materials in this chapter are taken from Chapter 11 of [1]. Additional theoretical information can be found in the same source.

2. Weapon Effectiveness of Unguided Weapons

Similar to the computation for single weapon unitary target, the two types of kill modes are blast and fragmentation. The weapon effectiveness of the weapon for area targets are also computed as a function of the accuracy function and the damage function shown in the form below. However, the main difference from the previous module is that the accuracy function is represented by expected fractional coverage of the weapon ($E(F_c)$), while the damage function is represented by the conditional damage probability (P_{CD}).

$$\mathrm{EFD} = \mathrm{E}(\mathrm{F_C}) * \mathrm{P_{CD}} \tag{5.18}$$

<u>LET/WET Computation for Fragmentation Kill Mode</u>

$$\text{Impact Angle Parameter, } \mathrm{a} = \max(1 - 0.8\cos(\text{Impact Angle}), 0.3) \tag{5.19}$$

$$\mathrm{LET} = \sqrt{\mathrm{MAE} * \mathrm{a}} \tag{5.20}$$

$$\mathrm{WET} = \frac{\mathrm{LET}}{\mathrm{a}} \tag{5.21}$$

There is a difference in the computation of LET as compared to the single weapon unitary target module as the damage function for area targets is in the form of a rectangle cookie cutter instead of a Carleton damage function.

<u>LET/WET Computation for Blast Kill Mode</u>

$$\mathrm{LET} = \mathrm{WET} = \sqrt{\mathrm{MAE}} \tag{5.22}$$

P_{CD} Computation

$$\text{Enlarged Weapon Lethal Length, } \mathrm{LEP} = \max(\mathrm{LET}, \mathrm{LA}) \quad (5.23)$$

$$\text{Enlarged Weapon Lethal Width, } \mathrm{WEP} = \max(\mathrm{WET}, \mathrm{WA}) \quad (5.24)$$

Note that LA and WA are the Length and Width of the target, which are manual inputs from the user.

$$\mathrm{P_{CD}} = \frac{\mathrm{LET} * \mathrm{WET}}{\mathrm{LEP} * \mathrm{WEP}} \quad (5.25)$$

Hence, if both the lethal length and width of the weapon are larger than the target dimensions, P_{CD} will be equal to one. Otherwise, it will be less than one as the weapon is not sufficiently lethal to damage the entire area of target elements.

E (F_c) Computation – Miss Distance Deviation Computation

$$\sigma_x = \mathrm{REP}/0.6745 \quad (5.26)$$

$$\sigma_y = \mathrm{DEP}/0.6745 \quad (5.27)$$

E (F_c) Computation – Integral Limits Computation

$$s_{range} = \frac{LEP+LA}{2} \quad (5.28), \qquad t_{range} = \frac{LEP-LA}{2} \quad (5.29)$$

$$a_{range} = \frac{s_{range}}{\sigma_x\sqrt{2}} \quad (5.30), \qquad b_{range} = \frac{t_{range}}{\sigma_x\sqrt{2}} \quad (5.31)$$

$$s_{deflection} = \frac{LEP+LA}{2} \quad (5.32), \qquad t_{deflection} = \frac{LEP-LA}{2} \quad (5.33)$$

$$a_{deflection} = \frac{s_{range}}{\sigma_x\sqrt{2}} \quad (5.34), \qquad b_{deflection} = \frac{t_{range}}{\sigma_x\sqrt{2}} \quad (5.35)$$

E (F_c) Computation – Integral Computation

$$I_{1\,(range)} = normcdf\left(t_{range}, 0, \sigma_x\right) - normcdf\left(-t_{range}, 0, \sigma_x\right) \quad (5.36)$$

$$I_{2\,(range)} = \left(\frac{LEP+LA}{2*LA}\right)\left[normcdf\left(t_{range}, 0, \sigma_x\right) - normcdf\left(-s_{range}, 0, \sigma_x\right)\right]$$

(5.37)

$$I_{3\,(range)} = \left(\frac{LEP+LA}{2*LA}\right)\left[normcdf\left(s_{range}, 0, \sigma_x\right) - normcdf\left(t_{range}, 0, \sigma_x\right)\right]$$

(5.38)

$$I_{4\,(range)} - I_{5\,(range)} = \frac{2\sigma_x}{L_A\sqrt{2\Pi}}\left(e^{-a^2} - e^{-b^2}\right) \quad (5.39)$$

$$I_{1\,(deflection)} = normcdf\left(t_{deflection}, 0, \sigma_y\right) - normcdf\left(-t_{deflection}, 0, \sigma_y\right) \quad (5.40)$$

$$I_{2\,(deflection)} = \left(\frac{WEP+WA}{2*WA}\right)\left[normcdf\left(t_{deflection}, 0, \sigma_y\right) - normcdf\left(-s_{deflection}, 0, \sigma_y\right)\right] \quad (5.41)$$

$$I_{3\,(deflection)} = \left(\frac{WEP+WA}{2*WA}\right)\left[normcdf\left(s_{deflection}, 0, \sigma_y\right) - normcdf\left(t_{deflection}, 0, \sigma_y\right)\right] \quad (5.42)$$

$$I_{4\,(deflection)} - I_{5\,(deflection)} = \frac{2\sigma_y}{W_A\sqrt{2\Pi}}\left(e^{-a^2} - e^{-b^2}\right) \quad (5.43)$$

<u>E (F_c) Computation</u>

$$E(F_{Range}) = I_{1\,(range)} + I_{2\,(range)} + I_{3(range)} + I_{4\,(range)} - I_{5\,(range)} \quad (5.44)$$

$$E(F_{deflection}) = I_{1\,(deflection)} + I_{2\,(deflection)} + I_{3(deflection)} + I_{4\,(deflection)} - I_{5\,(deflection)} \quad (5.45)$$

$$E(F_c) = E(F_{Range}) * E(F_{deflection}) \quad (5.46)$$

<u>EFD Computation</u>

$$EFD = E(F_C) * P_{CD} * Weapons\ Reliability \quad (5.47)$$

3. Weapon Effectiveness of Guided Weapons

The methodology to compute the weapon effectiveness of guided weapons for area targets is similar to the one for unitary target, which is to compute an EFD_2 and set the miss distance deviation to zero. Otherwise, the form is the same as follows:

$$EFD = EFD1 * P_{NM} + EFD2 * P_{HIT} \quad (5.48)$$

4. Single Weapon Area Target Module in MATLAB

The single weapon area target module is modeled as a function using the equations explained above. The input commands and the associated outputs for the module are depicted in Table 20. A flowchart of the module is shown in Figure 17.

Input	`Single_Wpn_Area_Target_Model( MAE, Kill_Mode, REP, DEP, Impact_Angle, Weapon_Reliability, Weapon_Type, LA, WA, P_Near_Miss, P_Hit)`
Output for Unguided Weapons	`Single_Wpn_Area_Target_output = [LET, WET, EF_R, EF_D, EFC, EFD];`
Output for Guided Weapons	`Single_Wpn_Area_Target_output = [LET, WET, EF_R, EF_D, EFC, EFD_1, EF_R_2, EF_D_2, EFC_2, EFD_2, EFD];`

Table 20. Input and output commands for single weapon area target module.

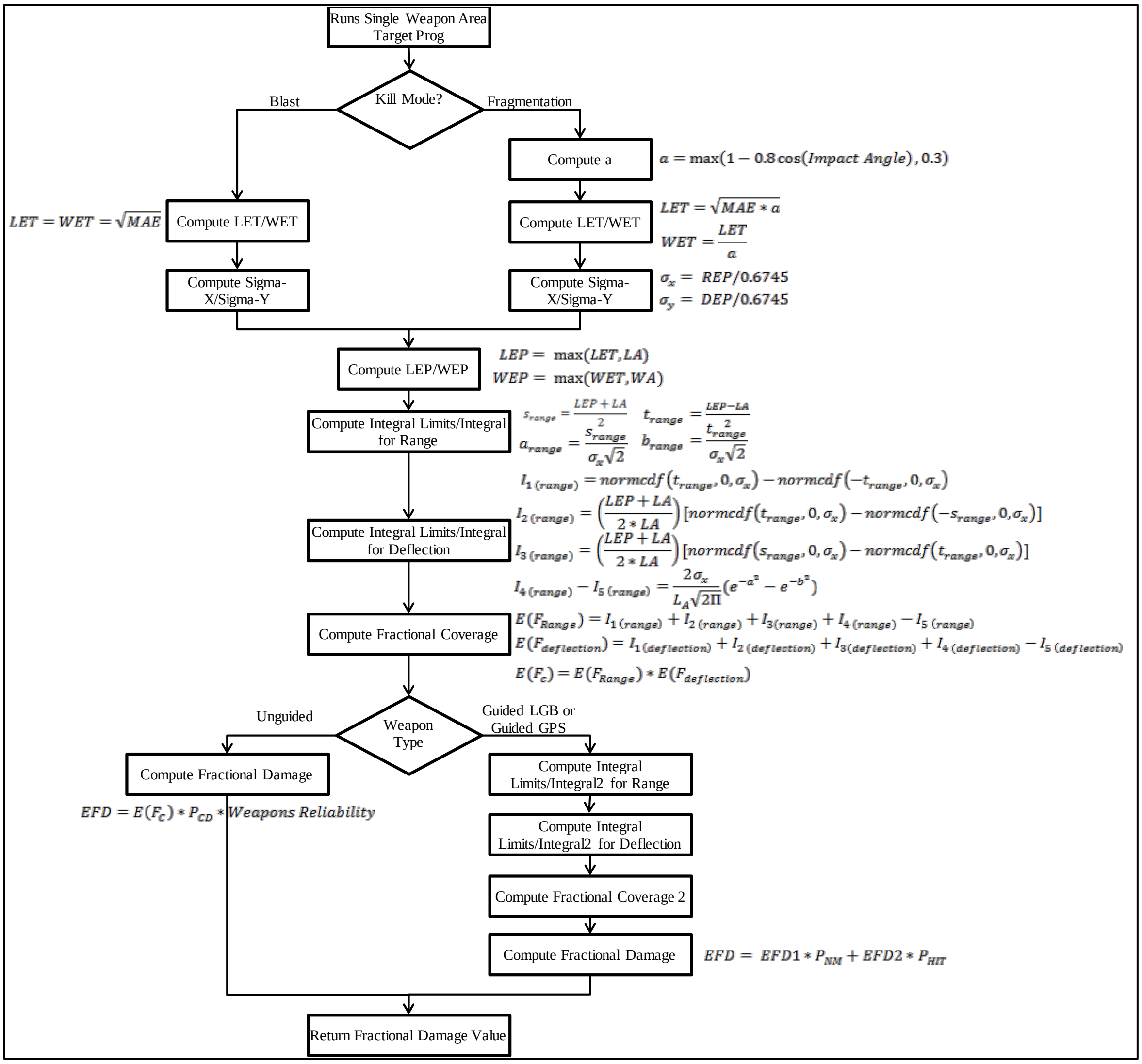


Figure 17. Flow diagram of single weapon area target module.

5. Results of Single Weapon Area Target Program

The results of the single weapon area target program were compared with an Excel program written by Prof. Driels [4] and were verified to be good. A print-screen of the result provided by Prof. Driels is illustrated in Figure 18. A print-screen of the result from the single weapon area target program is illustrated in Figure 19.

Target area dimensions	LA	200.000				
	WA	80.000				
WEAPON PARAMETERS						
Drag - L=low, H=high		l				
Ballistic dispersion (mils)	Sigma-b	5.000				
Reliability	R	0.680				
AIRCRAFT RELEASE DATA						
Airspeed -KTAS / ft/sec	V	500.000		844.000		
Dive angle, degrees	theta	30.000		0.524		
Release altitude, (ft)	yi	3500.000				
DELIVERY ACCURACY						
Adjusted DEP	DEP_dash	31.81				
Adjusted REP	REP_dash	49.55				
WEAPON EFFECTIVENESS						
Effectiveness index type			MAE_b	MAE_f		
Effectiveness value	EI		0.000	10000		
Length/width ratio (frag only)	a	0.673				WET/LET
Effective target width ft	WET		0.00	121.90		121.90
Effective target length ft	LET		0.00	82.03		82.03
Effective target area ft^2	AET		0	10000		10000
FD_1 CALCULATION						
	RANGE			DEFLECTION		
Sigma range (ft)	73.468			47.160		
LEP	200.000		WEP	121.905		
INTEGRAL LIMITS						
s	200.000		s	100.952		
t	0.000		t	20.952		
a	1.925		a	1.514		
b	0.000		b	0.314		
INTEGRALS						
integral #1	0.000		integral #1	-0.523		
integral #2	0.497		integral #2	0.827		
integral #3	0.497		integral #3	0.827		
integral #4-integral#5	-0.286		integral #4	-0.379		
RESULTS						
Fractional coverage	0.708			0.753		
Combined fractional coverage	0.533					
FD_1	**0.149**					

Figure 18. Results from Prof. Driels' Excel program. From [4]

Trajectory Results				
Vertical Velocity	ft/s	421.95		
Horizontal Velocity	ft/s	730.8388		
Drag Constant		0.25		
				Dragless
TOF	s	12.059	-	6.622658
Horizontal Velocity at Impact	ft/s	124.5843	-	730.8388
Vertical Velocity at Impact	ft/s	278.0771	-	635.0274
Impact Angle	deg	65.86663	-	40.98745
Impact Velocity	ft/s	304.7099	-	968.1866
Slant Range	ft	4990.412	-	5972.983
Ground Range	ft	3557.276	-	4840.096
δR/δH		1.135242	-	-
δR/δVx		6.622658	-	-
δR/δVz		-7.21844	-	-
		-	-	-
		-	-	-
		-	-	-
		-	-	-

Delivery Accuracy		
CEP		
DEP		27
REP		46
Ballistic error, normal plane	mils	5
Ballistic error, deflection	ft	16.83016
Ballistic error, range	ft	18.44207
Adjusted DEP	ft	31.81595
Adjusted REP	ft	49.55916

Weapon Effectiveness			
Effectiveness Index Type		MAE_b	MAE_f
Effectiveness Value	EI		10000
Length/Width Ratio (Frag Only)	a		0.672910362
Effective Target Width, ft	WET		121.9049644
Effective Target Length, ft	LET		82.03111371
Effective Target Area, ft^2	AET		107639.1042
Cond Prob of Damage	PCD		0.410155569

EFD Calculation		Range	Deflection
Sigma (ft)		73.47539901	47.16967572
LEP/WEP		200	121.9049644
INTEGRAL LIMITS			
s		200	100.9524822
t		0	20.95248222
a		1.924744311	1.513349067
b		0	0.314092519
INTEGRALS			
I1		0	0.343097598
I2		0.496755588	0.394070346
I3		0.496755588	0.394070346
I4-I5		-0.285910916	-0.37862356
RESULTS			
Fractional Coverage		0.70760026	0.752614735
Combined Fractional Coverage		0.532550382	
EFD		0.148531383	

Figure 19. Results from single weapon area target program.

C. CALLING OF SINGLE WEAPON UNITARY TARGET FUNCTION AND SINGLE WEAPON AREA TARGET FUNCTION

As the single weapon unitary target function and single weapon area target function shares similar input requirements for unguided, LGB and GPS guided weapons, the Main User Interface module computes and retrieves the required input requirements before deciding which of the functions to call. A flowchart of an overview of the program logic is shown in Figure 20. The flowchart of the program logic for unguided weapons is shown in Figure 21. The flowchart of the program logic for LGB is shown in Figure 22. The flowchart of the program logic for GPS guided weapons is shown in Figure 23.

Note also the following:

1) Impact Angle Input: There are different ways in which the impact angle input can be provided to the module, depending on the type of weapons being selected: unguided, LGB or GPS guided.

   a. Unguided: The impact angle of unguided weapons is taken from the high-fidelity trajectory program.

   b. LGB: There are three methods to provide the impact angle. The first is through manual input. The second method is to take the impact angle from the high-fidelity trajectory module to cater for the scenario that the targeting pod is only turn-on during the final phase of the weapon's flight for minor corrections to minimize the possibility of laser detection warning received by the target. The last method is to take the harp angle of the weapon as the impact angle to cater for the scenario where the weapon flies directly to the target upon weapon release as laser guidance is available throughout its flight. The harp angle is the line-of-sight (LOS) angle between the aircraft and the target at the point of weapon release and can be computed as follows:

$$Harp\ Angle = \ tan^{-1}(\frac{Release\ Altitude}{Distance\ to\ Target\ at\ Point\ of\ Weapon\ Release}) \qquad (5.49)$$

   c. GPS Guided: The impact angle for GPS guided weapons is based on the impact angle input by the user.

2) REP/DEP computations: The methodology of the REP/DEP computations was covered in previous chapters.

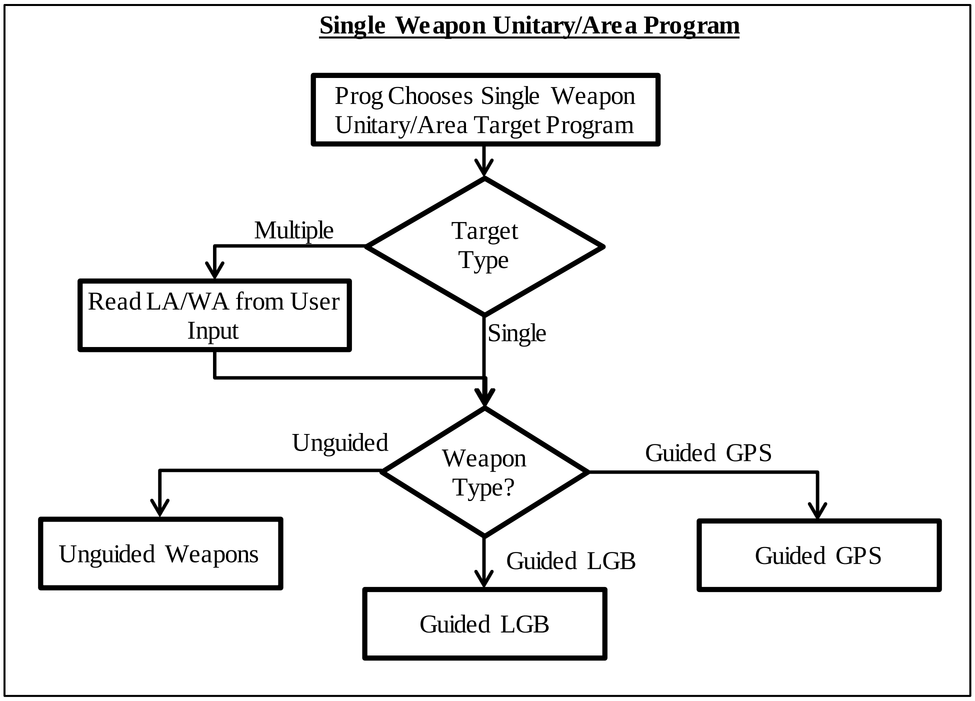


Figure 20. Overview of program logic to call single weapon unitary/area target function.

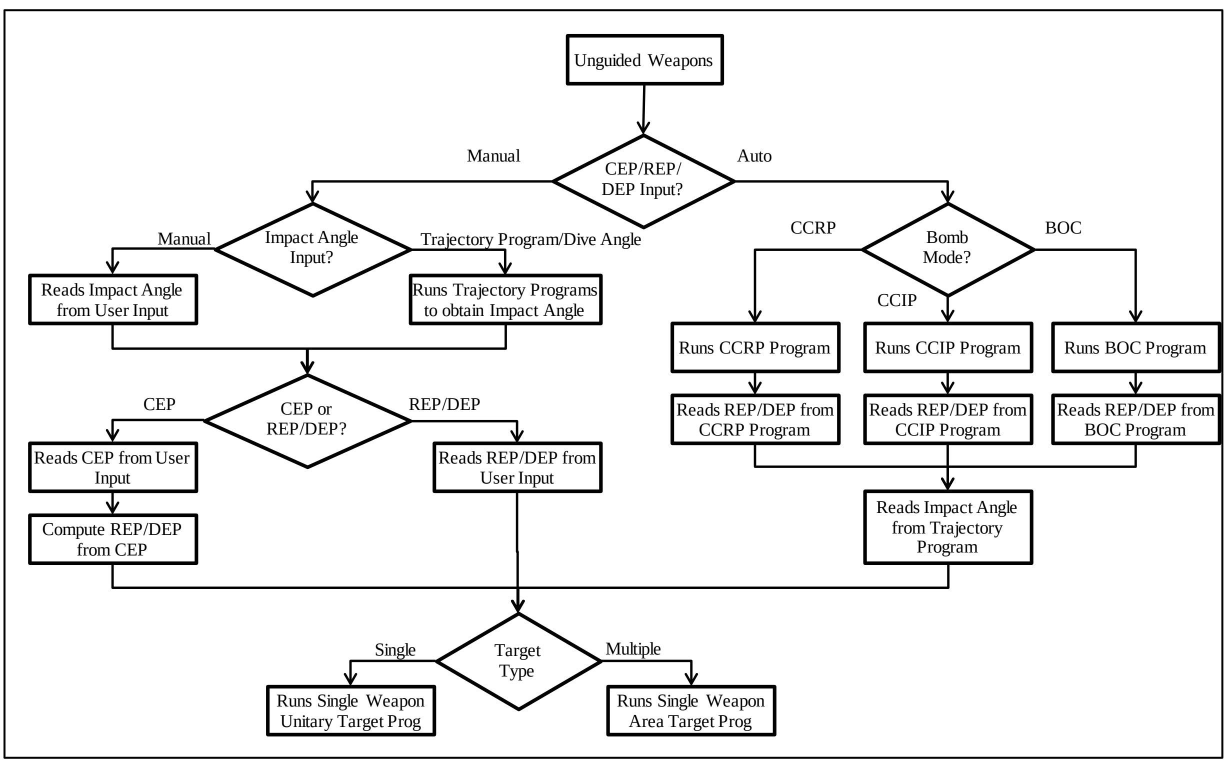


Figure 21. Program logic to call single weapon unitary/area target function for unguided weapons.

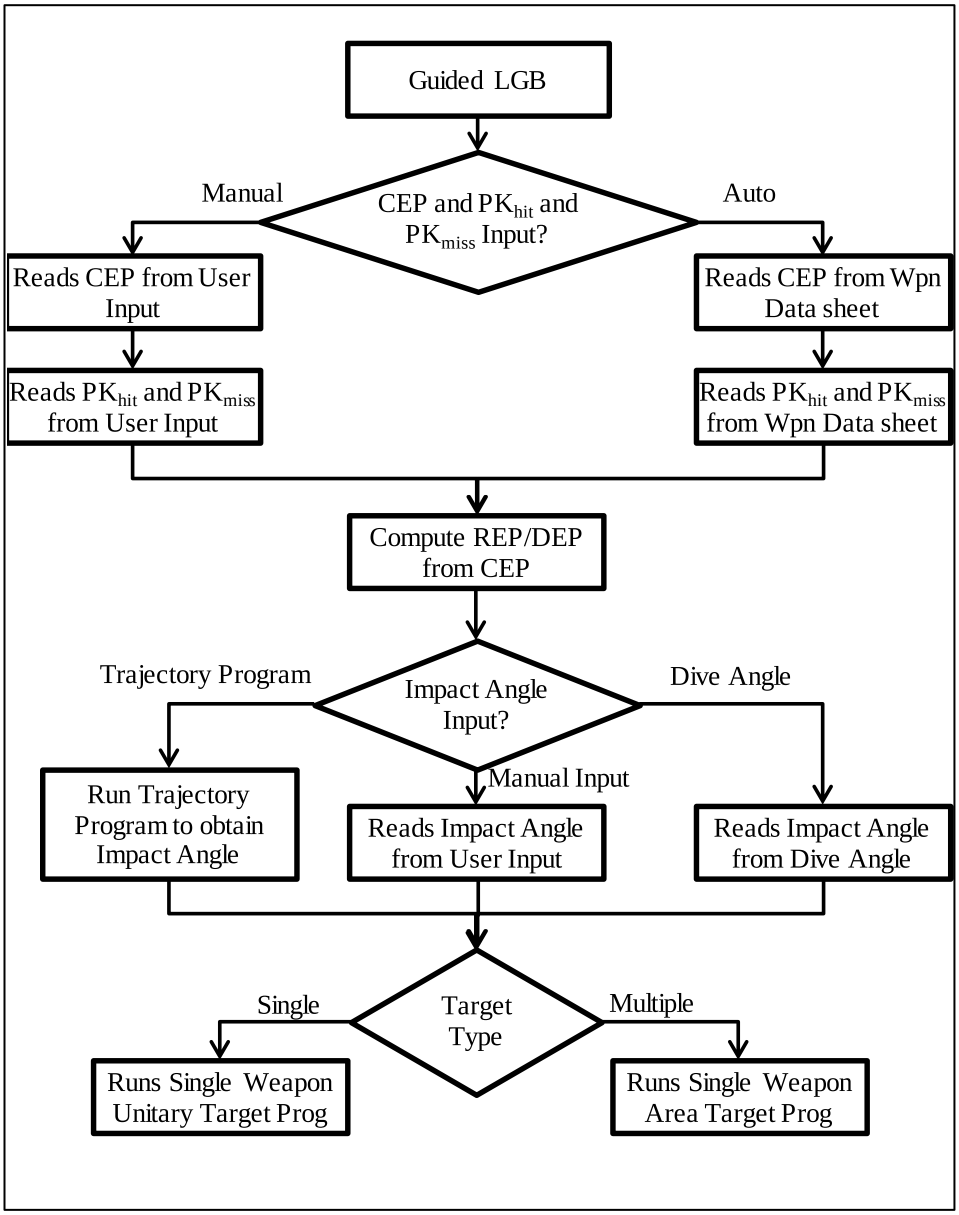


Figure 22. Program logic to call single weapon unitary/area target function for LGB.

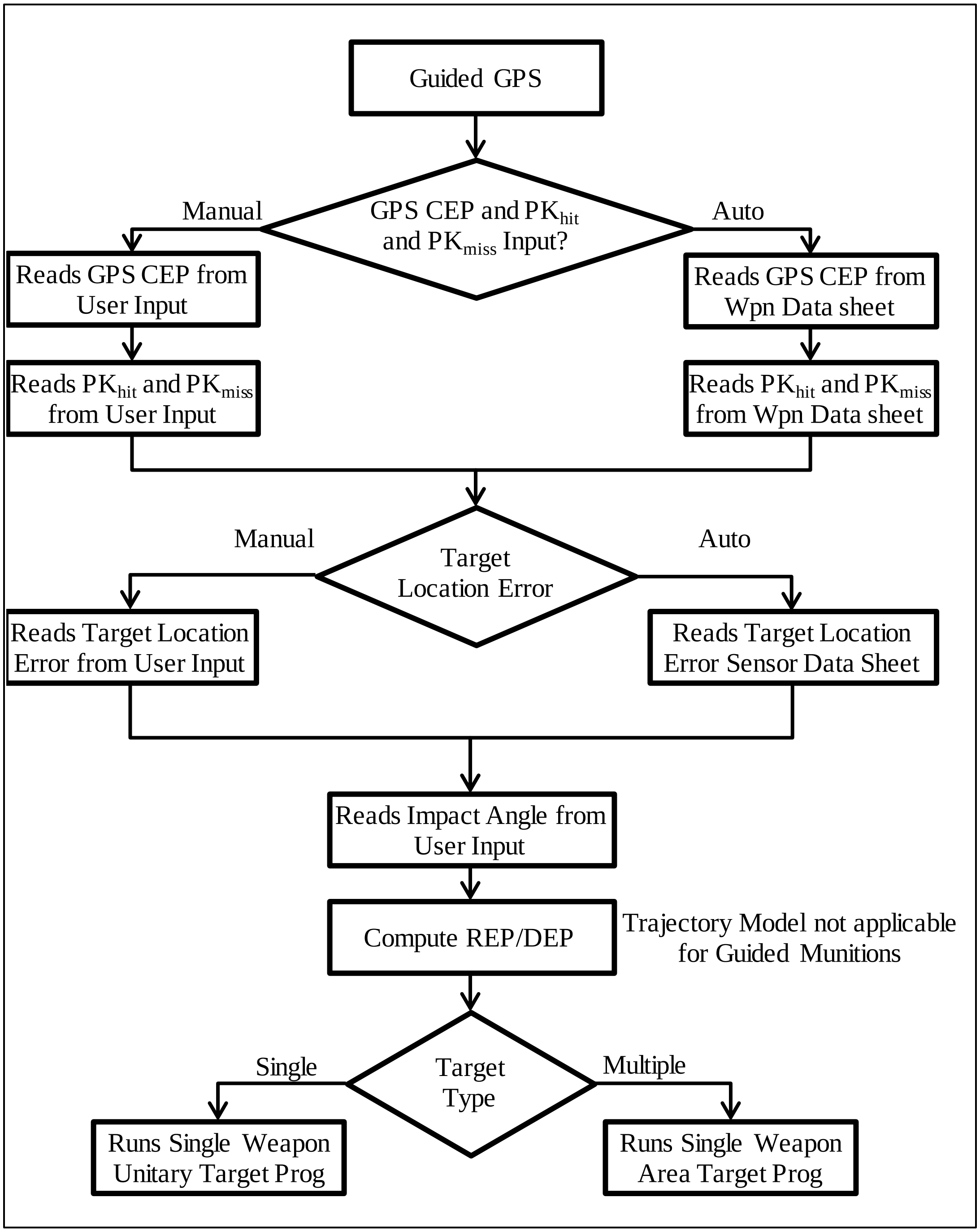


Figure 23. Program logic to call single weapon unitary/area target function for GPS guided weapons.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. STICK DELIVERY PROGRAM

A. BACKGROUND

The stick delivery program computes the probability of damage from employing multiple AS weapons against an area target. This probability of damage is being represented as the Expected Fractional Damage (EFD). The program only looks into the weapon effectiveness of unguided weapons. The materials in this chapter are taken from Chapter 13 of [1]. Additional theoretical information can be found in the same source.

B. COMPUTATION OF STICK DELIVERY WEAPON EFFECTIVENESS

The methodology to compute the weapons effectiveness for stick delivery is similar to that for a single weapon area target as both programs study the effectiveness of the weapon against area targets. There are additional computations required for stick delivery as there is more than one weapon being dropped against the target as illustrated in Figure 24. There is a need to consider the lethal area of all the weapons, including the effect of ballistic dispersion, and look at the effect of the entire pattern area. Hence, the computation of EFD for stick delivery takes the form of:

$$EFD = E(F_C) * \frac{A_P}{L_{EP} * W_{EP}} * R * P_{CDS} \tag{6.1}$$

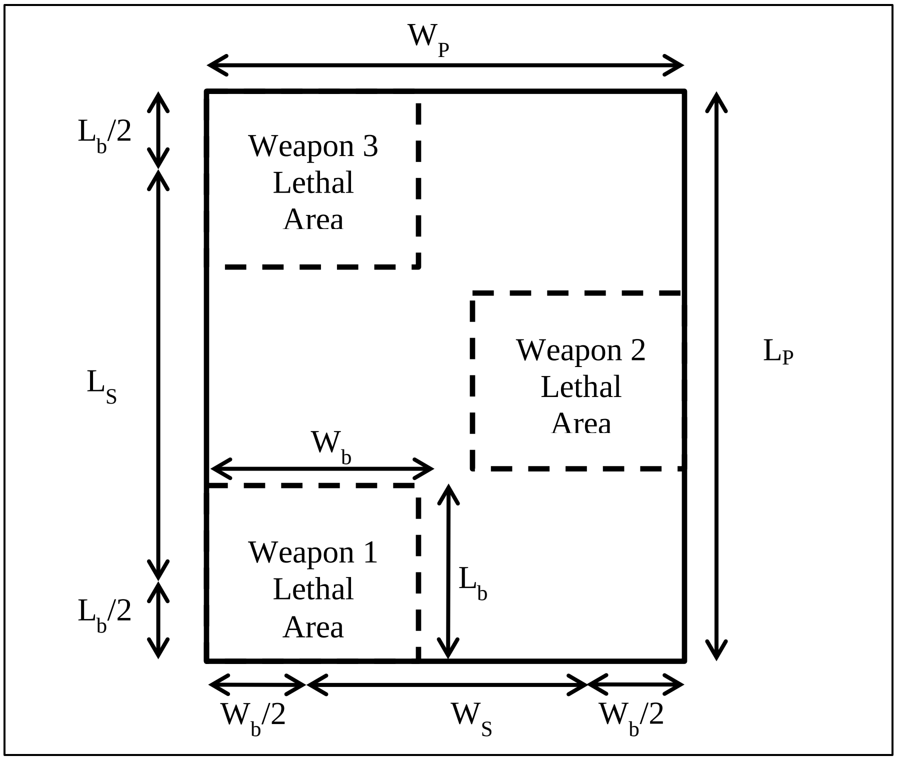


Figure 24. Illustration of stick delivery.

Stick Length Computation

While stick width is a user-input based on the distance between the weapon stations onboard the aircraft, stick length has to be computed by the program based on the intervalometer setting and the release condition of the aircraft. For the program, users can define an intervalometer setting in distance or time if the aircraft is flying straight and level. Users can only defined a time setting if the aircraft is flying at a non-zero dive angle (i.e., in a dive).

Stick Length Computation for Time Setting at Zero Dive Angle:

$$L_S = \text{Aircraft Velocity} * (n_r - 1) * \text{IV Time} \tag{6.2}$$

Stick Length Computation for Distance Setting at Zero Dive Angle:

$$L_S = (n_r - 1) * \text{IV Distance} \tag{6.3}$$

Stick Length Computation for Time Setting at Non-Zero Dive Angle:

$$\mathrm{L_S = Aircraft - x\,Velocity * (n_r - 1) * IV\,Time + R_B\,of\,Last\,Weapon - R_B\,of\,First\,Weapon} \tag{6.4}$$

The value of R_B has to be computed based on the high fidelity trajectory program for sticks, which will be elaborated on subsequently.

<u>LET/WET Computation</u>

Fragmentation Kill Mode:

$$\textit{Impact Angle Parameter},$$
$$\mathrm{a = max(1 - 0.8\cos(Average\ Impact\ Angle)\,, 0.3)} \tag{6.5}$$

$$\mathrm{LET} = \sqrt{\mathrm{MAE * a}} \tag{6.6}$$

$$\mathrm{WET} = \frac{\mathrm{LET}}{\mathrm{a}} \tag{6.7}$$

Note that the impact angle is generated by the high-fidelity trajectory module and is computed as the average impact angle of the first released weapon and the last released weapon as shown:

$$\mathrm{Average\ Impact\ Angle} = \frac{\mathrm{Impact\ Angle_{First\ Weapon} + Impact\ Angle_{Second\ Weapon}}}{2} \tag{6.8}$$

Blast Kill Mode:

$$\mathrm{LET = WET} = \sqrt{\mathrm{MAE}} \tag{6.9}$$

<u>Ballistic Dispersion Errors Computation (σ_{br} and σ_{bd})</u>

The ballistic dispersion errors in range (σ_{br}) and deflection (σ_{bd}) are computed based on the outputs of the high-fidelity trajectory module as follows:

$$\mathrm{Average\ Release\ Altitude} = \frac{\mathrm{First\ Release\ Altitude + Final\ Release\ Altitude}}{2} \tag{6.10}$$

$$\mathrm{Ground\ Range\ to\ Pattern\ Centre = R_B\ offirst\ weapon} + \frac{\mathrm{L_S}}{2} \tag{6.11}$$

$$\textit{Slant Range to Pattern Centre} = \sqrt{\mathrm{Average\ Release\ Altitude^2 + Ground\ Range\ to\ Pattern\ Centre^2}} \tag{6.12}$$

$$\sigma_{\mathrm{bd}} = \frac{\mathrm{Slant\ Range\ to\ Pattern\ Centre} * \sigma_{\mathrm{b}}}{1000} \tag{6.13}$$

$$\sigma_{br} = \frac{\text{Slant Range to Pattern Centre}*\sigma_b}{1000*\sin(\text{Impact Angle of First Weapon})} \tag{6.14}$$

<u>L_B/W_B Computation</u>

$$L_B = \sqrt{LET^2 + 8\sigma_{br}{}^2} \tag{6.15}$$

$$W_B = \sqrt{WET^2 + 8\sigma_{bd}{}^2} \tag{6.16}$$

$$P_{CD1} = \frac{LET*WET}{L_B*W_B} \tag{6.17}$$

<u>Computing Pattern Dimensions L_P and W_P</u>

$$L_P = L_S + L_B \tag{6.18}$$

$$W_P = W_S + W_B \tag{6.19}$$

<u>Possible Overlap in Deflection Direction</u>

$$n_{od} = \frac{n_p W_B}{W_P} \tag{6.20}$$

If $n_{od} \geq 1$, where the weapon effectiveness area overlaps,

$$P_{CD/d} = 1 - (1 - P_{CD1})^{n_{od}} \tag{6.21}$$

If $n_{od} < 1$, where the weapon effectiveness area does not overlaps,

$$P_{CD/d} = n_p P_{CD1} \frac{W_B}{W_P} \tag{6.22}$$

<u>Possible Overlap in Range Direction</u>

$$n_{or} = \frac{n_r L_B}{L_P} \tag{6.23}$$

If $n_{od} \geq 1$, where the weapon effectiveness area overlaps,

$$P_{CD/S} = 1 - (1 - P_{CD/d})^{n_{or}} \tag{6.24}$$

If $n_{od} < 1$, where the weapon effectiveness area does not overlaps,

$$P_{CDS} = n_r P_{CD/d} \frac{L_B}{L_P} \tag{6.25}$$

$$L_{EP} = \max(L_P, L_A) \tag{6.26}$$

$$W_{EP} = \max(W_P, W_A) \tag{6.27}$$

Computation of $E(F_C)$

The methodology to compute $E(F_C)$ is exactly the same as the one used for the single weapon area target module and will not be covered in this chapter. The exception is that the fractional coverage is computed by comparing the pattern area of the sticks of a weapon with the target area instead of using the individual weapon lethal area represented by LET and WET. Hence, the LEP equation above is used instead of the equation used in the single weapon area target module.

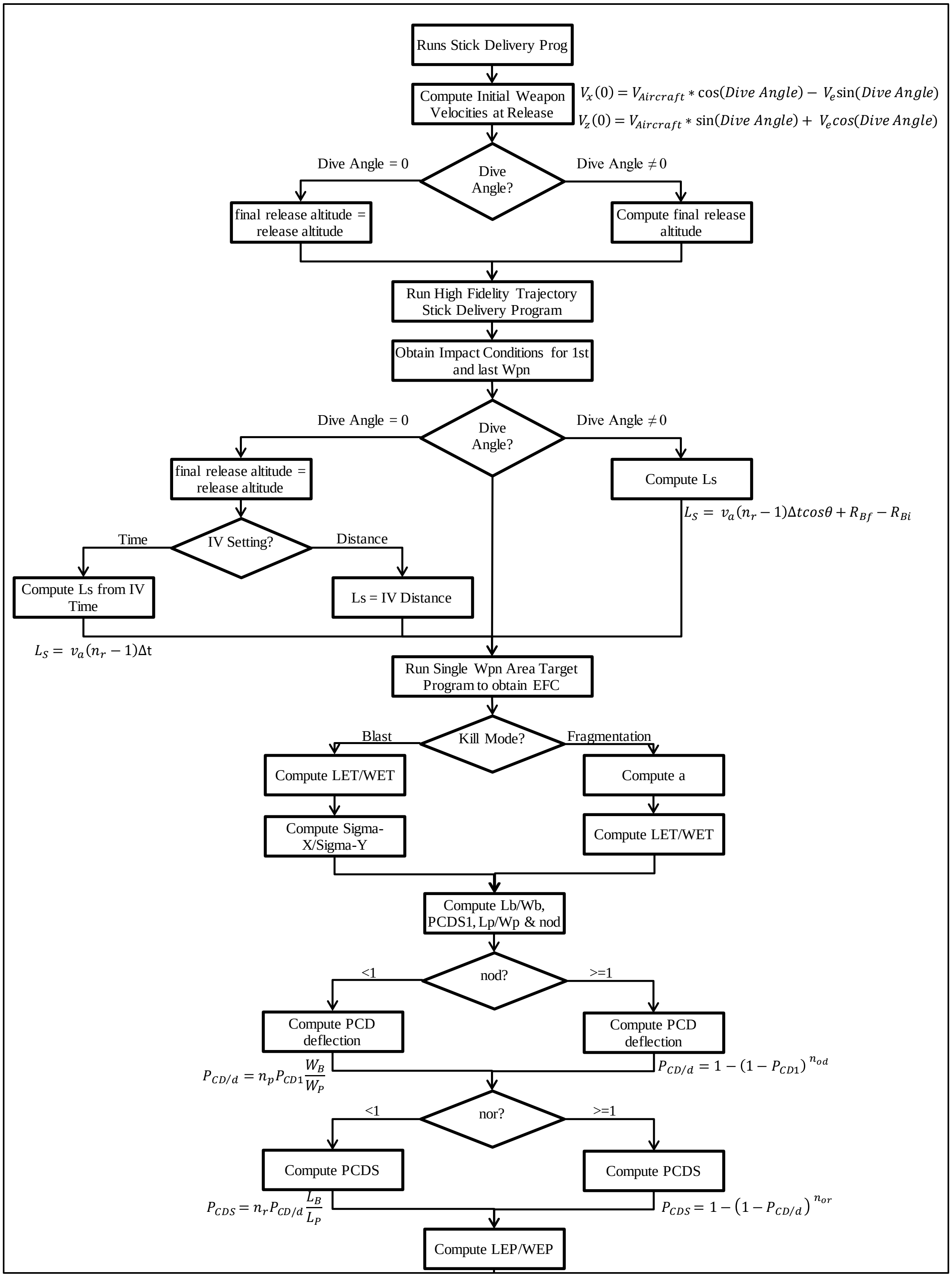
$$EFD = E(F_C) * \frac{A_p}{L_{EP} * W_{EP}} * R * P_{CDS} \tag{6.28}$$

C. STICK DELIVERY MODULE IN MATLAB

The stick delivery module is modeled as a function using the equations explained in the previous section. The input commands and the associated outputs for the module are depicted in Table 21. A flowchart of the module is shown in Figure 25.

Input	`Stick_Delivery_Model(release_altitude, dive_angle, ac_velocity, ejection_velocity, MAE, Kill_Mode, REP, DEP, Weapon_Reliability, Weapon_Type, Warhead_Size, LA, WA, nr, np, IV_Setting, IV_Time, IV_distance, Ws, Sigma_b)`
Output	`Stick_Delivery_output = EFD;`

Table 21. Input and output commands for stick delivery module.



(Continued on next page)

(Continued from previous page)

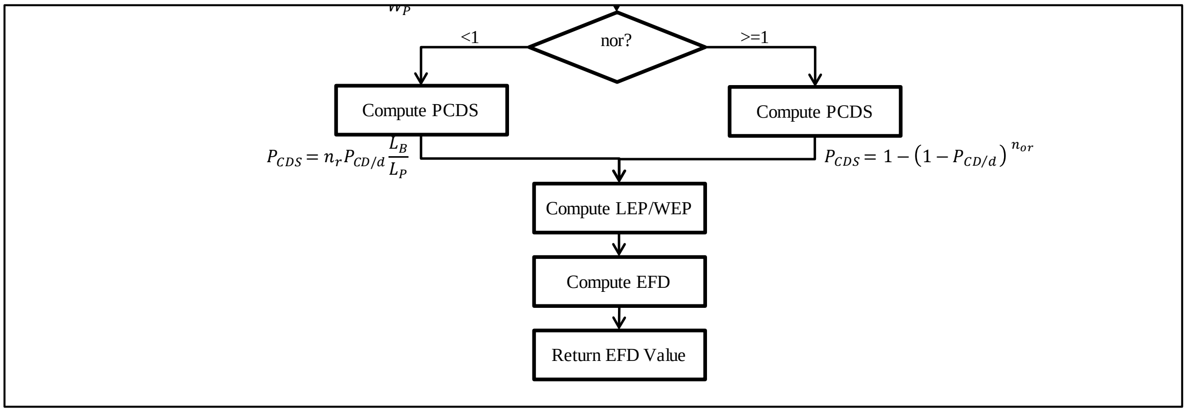


Figure 25. Flow diagram of stick delivery module.

Note also the following:

1) Single Weapon on Parent Rack: Due to the complexity of the programming required, the program assumes that there is only one weapon on each parent rack. The reason for this is because weapon release from the shoulder stations of armament role equipment such as a Triple-Ejection-Rack (TER) would have resulted in an angle between the fuselage and the ejection velocity vector, which is difficult to program.

2) There is a maximum of two weapons released per pulse as the theory in [1] is derived for such a scenario only.

3) Intervalometer setting in distance is only available for straight and level flight profile.

D. RESULTS OF STICK DELIVERY MODULE IN MATLAB

The results of the stick delivery program were compared with an Excel program provided by Prof. Driels [5] and were verified to be good. A print-screen of the result provided by Prof. Driels is illustrated in Figure 26. A print-screen of the result from the stick delivery program is illustrated in Figure 19.

TARGET PARAMETERS						
Target length	LA	1000.000				
Target width	WA	1300.000				
WEAPON PARAMETERS						
Weapon-L=low drag, H=high drag		H				
Ballistic dispersion (mils)	Sigma-b	10.000				
Reliability	R	0.950				
Airspeed -KTAS / ft/sec	V	500.000		844.000	ft/sec	
Dive angle, degrees	theta	10.000		0.175	radians	
# of weapons, external racks	ne	0.000				
# of weapons, internal racks	ni	0.000				
# of release pulses	nr	6.000				
# weapons/release pulse	np	2.000				
Intervalometer setting (sec)	dt	0.130				
Release alt. First weapon ft	yi	1132.000				
Release alt. Last weapon ft	yn	1036.737				
Average release altitude ft	ybar	1084.368				
Stick length, ft	Ls	307.8				
Stick width, ft	Ws	40.000				
Slant range, first weapon, ft	Sri	3829				
Ground range to pattern center, (ft)	Gr	3812				
Slant range to pattern center, (ft)	SR	3963				
Deflection error probable, ft	DEP	0.000	0	0.00	10	10.000
Range error probable, ft	REP	0.000	0	0.00	30	30.000
Ballistic error, deflection, ft	sigma-bd	39.634		10		
Ballistic error range, ft	sigma-br	87.301				
Effectiveness index type			MAE_b	MAE_f		
Effectiveness value	EI		0.000	23847.0		
Length/width ratio (frag only)	a	0.300				WET/LET
Effective target width ft	W_{ET}		0.000	281.9	281.9	281.9
Effective target length ft	L_{ET}		0.000	84.6	84.6	84.6
Effective target area ft^2	A_{ET}		0.000	23847.0		23847.0
Single weapon effective dim, ft	WB	303.4				
	LB	261.0				
	AB	79193				
Effective stick pattern, ft	WP	343.4				
	LP	568.8				
	AP	195334				
Conditional prob of damage	PCD_1	0.301				
Degree of overlap - deflection	No_d	1.767				
Cond prob damage-deflection	PCDd	0.469				
Degree of overlap - range	No_r	2.753				
Prob damage within pattern	PCDS	0.825				

FD_1 CALCULATION						
	RANGE		DEFLECTION			
Sigma (ft)	44.477		14.826			
Effective pattern size LEP/WEP	1000.000		1300.000			
INTEGRAL LIMITS						
s	1000.000		1300.000			
t	0.000		0.000			
a	15.898		-62.003			
b	0.000		0.000			
INTEGRALS						
integral #1	0.000		0.000			
integral #2	0.500		0.500			
integral #3	0.500		0.500			
integral #4 - integral #5	-0.035		-0.009			
Fractional coverage Fr/Fd	0.965		0.991			
FD_1	**0.113**					

Figure 26. Results from Prof. Driels' Excel program. From [5]

Trajectory Results				
Vertical Velocity	ft/s	146.5417		
Horizontal Velocity	ft/s	831.0793		
Drag Constant		0.11		
		1st Wpn	Last Wpn	Dragless
TOF	s	6.06	5.552	-
Horizontal Velocity at Impact	ft/s	448.219	467.2624	-
Vertical Velocity at Impact	ft/s	228.3649	221.3759	-
Impact Angle	deg	26.99857	25.35029	-
Impact Velocity	ft/s	503.0415	517.0508	-
Slant Range	ft	3829.402	3573.765	-
Ground Range	ft	3658.264	3425.797	-
δR/δH		-	-	-
δR/δVx		-	-	-
δR/δVz		-	-	-
Stick Length	ft	307.7344	-	-
Stick Width	ft	40	-	-
Ground Range to Pattern Center	ft	3812.131	-	-
Slant Range to Pattern Center	ft	3960.763	-	-

Delivery Accuracy		
CEP		
DEP		10
REP		30
Ballistic error, normal plane	mils	10
Ballistic error, deflection	ft	39.60763
Ballistic error, range	ft	87.24758
Adjusted DEP	ft	
Adjusted REP	ft	

Weapon Effectiveness						
Effectiveness Index Type		MAE_b	MAE_f			
Effectiveness Value	EI		23847			
Length/Width Ratio (Frag Only)	a		0.3			
Effective Target Width, ft	WET		281.9397099			
Effective Target Length, ft	LET		84.58191296			
Effective Target Area, ft^2	AET		23847			
Single Weapon Effective Dim ft	WB	303.381139		Effective Stick Pattern, ft	WP	343.3811
	LB	260.8662848			LP	568.6007
	AB	79141.9106			AP	195246.8
Conditional Prob of Damage	PCD 1	0.301319488		Degree of Overlap - Deflection	Nod	1.767023
Cond Prob Damage - Deflection	PCD-d	0.469315016		Degree of Overlap - Range	Nor	2.752718
Prob Damage within Pattern	PCDS	0.825195266				

EFD Calculation		Range	Deflection
Sigma (ft)		44.47739066	14.82579689
LEP/WEP		1000	1300
INTEGRAL LIMITS			
s		1000	1300
t		0	0
a		15.89811746	62.00265811
b		0	0
INTEGRALS			
I1		0	0
I2		0.5	0.5
I3		0.5	0.5
I4-I5		-0.035487823	-0.00909944
RESULTS			
Fractional Coverage		0.964512177	0.990900558
Combined Fractional Coverage		0.955735654	
EFD		0.112527487	

Figure 27. Results from stick delivery program.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CLUSTER PROGRAM

A. BACKGROUND

The cluster program computes the probability of damage from employing cluster weapons against an area target. This probability of damage is represented as the Expected Fractional Damage (EFD). The cluster weapon can be dropped as a single weapon or as a stick delivery. The materials in this chapter are taken from Chapter 15 of [1]. Additional theoretical information can be found in the same source.

B. COMPUTATION OF CLUSTER WEAPON EFFECTIVENESS

The methodology to compute the weapons effectiveness for cluster weapons is similar to that for a single weapon area target as both programs study the effectiveness of the weapon against area targets. When the cluster weapon is dropped as a stick delivery, the methodology follows closely that of the stick delivery module. The main difference is the additional computation required to account for the lethal area and ballistic dispersion of the numerous submunitions, which are released from the main cluster weapon at a preset time or altitude. Nevertheless, the EFD computation for cluster weapons takes the same form as stick delivery:

$$\mathrm{EFD} = \mathrm{E}(\mathrm{F_C}) * \frac{\mathrm{A_p}}{\mathrm{L_{EP}} * \mathrm{W_{EP}}} * \mathrm{R} * \mathrm{P_{CDS}} \qquad (7.1)$$

1. Effectiveness of Cluster Weapon as a Single Weapon

The damage function of the cluster weapon is dependent on whether the damage pattern of the bomblet is defined as a circular or rectangular pattern. This affects the computation of the expected number of submunitions that can damage the target given the target coverage by the damage pattern (n_x). The computation of edge effect (R_A), LET, WET and probability of damage by a single bomblet (P_s) are also dependent on the damage pattern. The dimension of a circular damage pattern is defined by the radius (R_P), while the dimension of a rectangular damage pattern is determined by its length (L) and width (W), all of which are user-defined parameters in the program.

Rectangular Damage Pattern

$$n_x = \frac{n_b * R_S * A'_{ET}}{L * W} \tag{7.2}$$

$$R_A = 2\sqrt{\frac{A'_{ET}}{\pi}}[1 - \exp(-0.18 * n_x)] \tag{7.3}$$

$$LET = L * 2R_a \tag{7.4}$$

$$WET = W * 2R_a \tag{7.5}$$

$$P_S = \frac{R_S * A'_{ET}}{L * W} \tag{7.6}$$

Circular Damage Pattern

$$n_x = \frac{n_b * R_S * A'_{ET} * \sin I}{\pi * {R_p}^2} \tag{7.7}$$

$$R_A = 2\sqrt{\frac{A'_{ET} * \sin I}{\pi}}[1 - \exp(-0.18 * n_x)] \tag{7.8}$$

$$WET = \sqrt{\pi}(R_P + R_A) \tag{7.9}$$

$$LET = \frac{WET}{\sin I} \tag{7.10}$$

$$P_S = \frac{R_S * A'_{ET}}{\pi * {R_p}^2} \tag{7.11}$$

Computing Probability of Damage of all Submunitions

Once P_s is computed, P_{CD} can be computed for all the submunitions in the cluster weapon using the survival rule.

$$P_S = 1 - (1 - P_S)^{n_b} \tag{7.12}$$

Expected Fractional Coverage (EFC) Computation

The methodology to compute EFC is the same as the one used for the single weapon area target module and will not be detailed here. There is a difference in the computation of the effective pattern dimensions (LEP and WEP) when the cluster weapon is dropped as a single weapon or stick delivery. The computation of LEP and WEP for single release of cluster weapon is the same as the one used for the single

weapon area target module as it compared the pattern of the single cluster weapon with the target area. In contrast, the computation of LEP and WEP for stick delivery of cluster weapons will follow that of the stick delivery module.

$$\text{Enlarged Weapon Lethal Length, } \text{LEP} = \max(\text{LET}, \text{LA}) \tag{7.13}$$

$$\text{Enlarged Weapon Lethal Width, } \text{WEP} = \max(\text{WET}, \text{WA}) \tag{7.14}$$

<u>Expected Fractional Damage (EFD) Computation</u>

The final EFD computation is similar to that used for the single release of a cluster weapon as follows:

$$\text{EFD} = \text{E}(\text{F}_\text{C}) * \frac{\text{LET}*\text{WET}}{\text{L}_\text{EP}*\text{W}_\text{EP}} * \text{R} * \text{P}_\text{S} \tag{7.15}$$

2. Effectiveness of Cluster Weapon as a Stick Delivery

The additional computation required to compute the effectiveness of the cluster weapons in stick delivery is exactly the same as the methodology used for stick delivery and will not be covered in detail here. The only difference is the slight difference in the computation of P_{CD1} as illustrated below, where P_s is not assumed to be 1.

$$\text{P}_{\text{CD1}} = \text{P}_\text{S} * \frac{\text{LET}*\text{WET}}{\text{L}_\text{B}*\text{W}_\text{B}} \tag{7.16}$$

In addition, as compared to the weapons effectiveness computation for a single cluster weapon, the LEP and WEP for the stick delivery of cluster weapons are computed as follows:

$$\text{Enlarged Weapon Lethal Length, } \text{LEP} = \max(\text{Lp}, \text{LA}) \tag{7.17}$$

$$\text{Enlarged Weapon Lethal Width, } \text{WEP} = \max(\text{Wp}, \text{WA}) \tag{7.18}$$

Hence, the final EFD computation for stick delivery of cluster weapons is as follows:

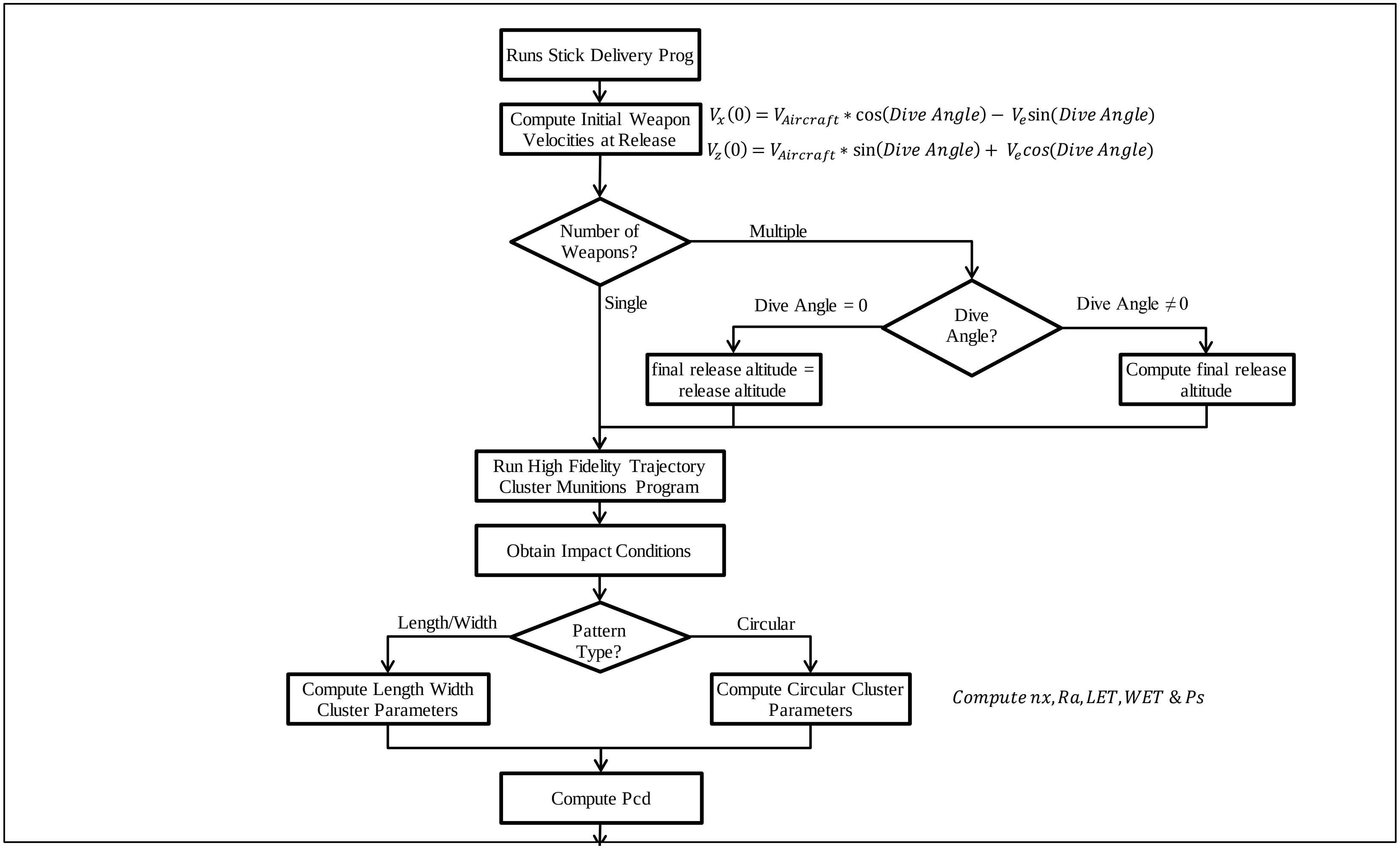
$$\text{EFD} = \text{E}(\text{F}_\text{C}) * \frac{\text{Lp}*\text{Wp}}{\text{L}_\text{EP}*\text{W}_\text{EP}} * \text{R} * \text{P}_\text{S} \tag{7.19}$$

C. CLUSTER WEAPONS MODULE IN MATLAB

The cluster weapons module is modeled as a function using the equations explained in the previous section. The input commands and the associated outputs for the module are depicted in Table 22. A flowchart of the module is shown in Figure 25.

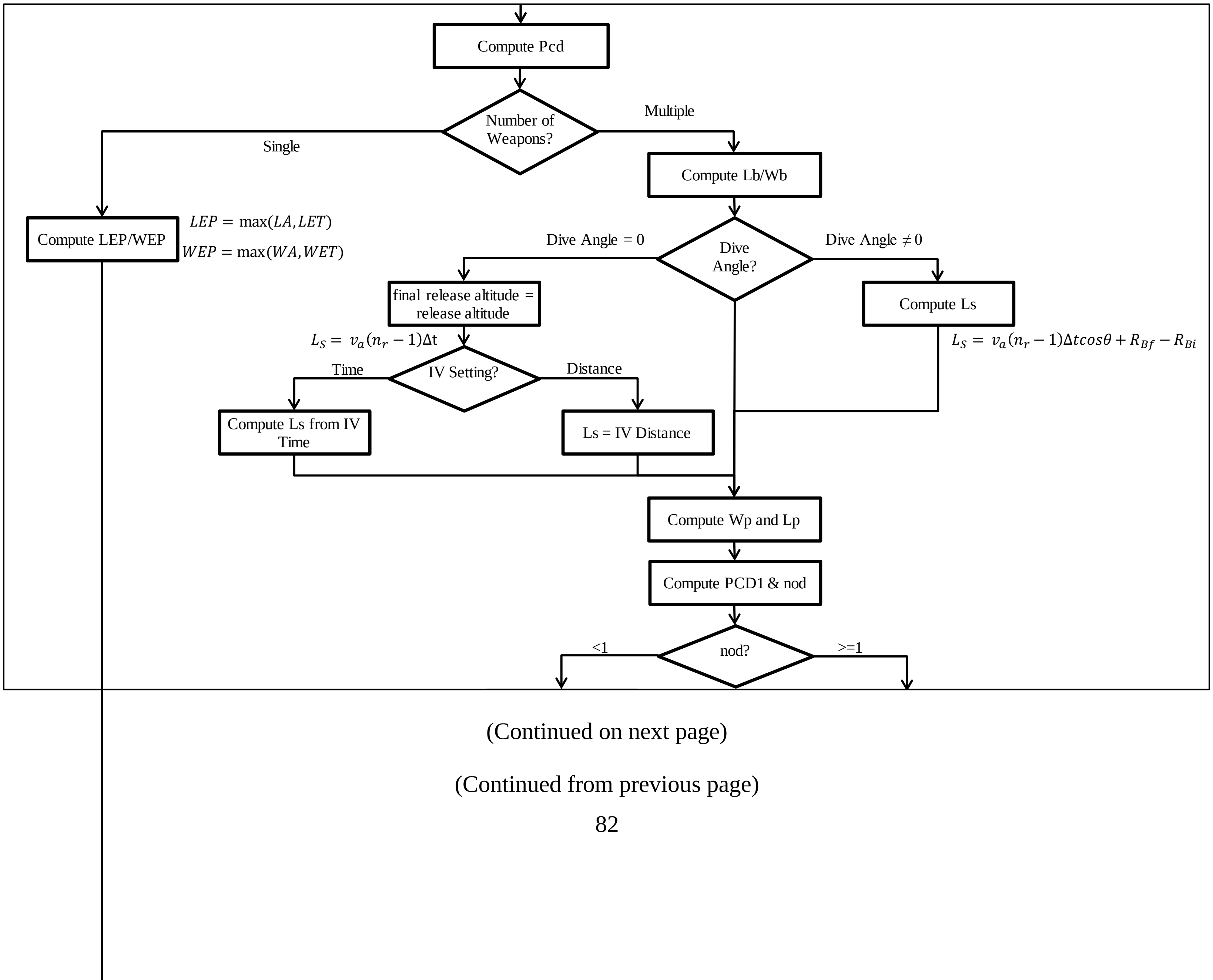
Input	`Cluster_Munitions_Model(release_altitude, Release_Type, functioning_time, functioning_altitude, nb, Rs, Pattern_Type, L, W, Rp, Number_of_Weapons, dive_angle, ac_velocity, ejection_velocity, MAE, Kill_Mode, REP, DEP, Dispenser_Reliability, Weapon_Type, Warhead_Size, LA, WA, nr, np, IV_Setting, IV_Time, IV_distance, Ws, sigma_b)`
Output	`Cluster_Munitions_output = EFD;`

Table 22. Input and output commands for cluster munitions module.



(Continued on next page)

(Continued from previous page)



(Continued on next page)

(Continued from previous page)

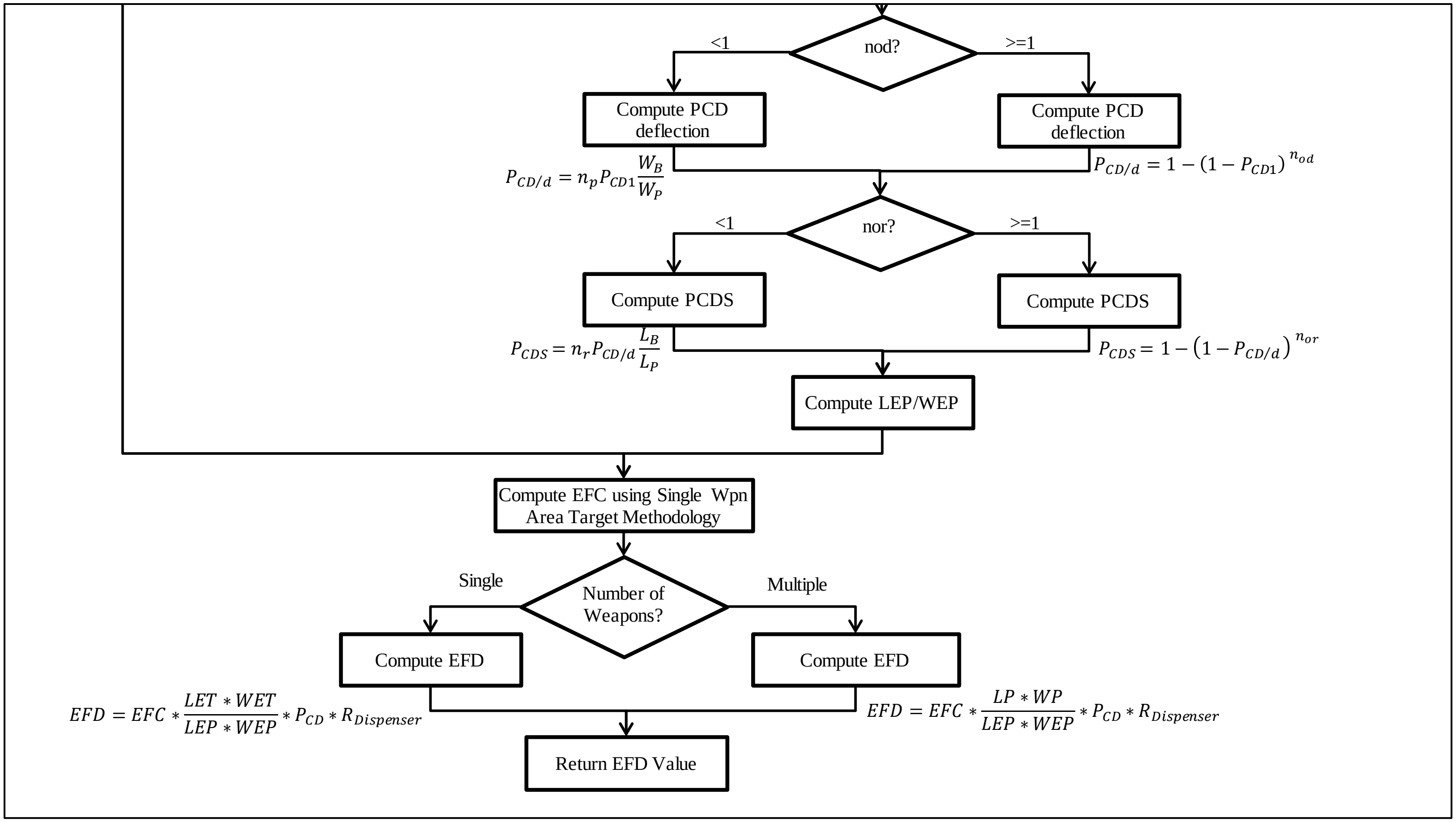


Figure 28. Flow diagram of cluster munitions module.

Note also the following:

1) For computation of LET and WET, there is no difference in the equations used between blast warheads and fragmentation warheads. The main differentiating factor is the MAE that is input into the system.

2) The pattern length/width or radius are assumed to be known by the user and are inputs to the module.

D. RESULTS OF CLUSTER MUNITIONS MODULE IN MATLAB

The results of the cluster munitions program were compared with an Excel program provided by Prof. Driels [6] and were verified to be good. A print-screen of the result provided by Prof. Driels is illustrated in Figure 29. Figure 26. A print-screen of the result from the cluster munitions program is illustrated in Figure 30. Figure 19.

TARGET PARAMETERS					
Target length, ft	LA	1000			
Target width, ft	WA	1000			
WEAPON PARAMETERS					
Number of submunitions/dispenser	n_b	202			
Ballistic dispersion (mils)	Sigma-b	10.200			
Functioning time (sec)	T_f	1.600			
Single dispenser pattern length (ft)	L	180.000			
Single dispenser pattern width (ft)	W	272.000			
Submunition reliability	R_b	0.930			
Dispenser reliability	R	0.800			
AIRCRAFT RELEASE DATA					
Airspeed -KTAS	V	550.000		928.400	ft/sec
# of release pulses	nr	6.000			
# weapons/release pulse	np	2.000			
Intervalometer setting (sec)	dt	0.130			
Release alt. First weapon ft	yi	1000.000			
Impact angle, degrees	I	31.643			
Weapon spacing on ground, ft	Sb	120.692			
Stick length, ft	Ls	603.460			
Stick width, ft	Ws	10.000		See Data Sheet	
Slant range, first weapon, ft	Sri	3830.889			
Ground range, first weapon, ft	Gri	3698.068			
Ground range, ft	Gr	3999.798			
Slant range, ft	SR	3830.888			
DELIVERY ACCURACY					
			mils	ground	REP/DEP
Deflection error probable, ft	DEP	150.000	0	0	150.000
Range error probable, ft	REP	170.000	0	0	170.000
WEAPON EFFECTIVENESS					
Effectiveness index type			MAE_b	MAE_f	AET
Effectiveness value-submunition	EI		0.000	4666.000	4666.000
# submunitions causing damage	nx	17.903			
Edge effect adjustment ft	RA	74.006			
Single cluster weapon					
Single dispenser width with RA	WET	420.0			
Single dispenser length with RA	LET	328.0			
Pd one submunition	Ps	0.089			
Pd all submunitions	Pcd_wep	1.000			
Single weapon effective width, ft	WB	434.3			
Single weapon effective length ft	LB	389.8			
Stick calculation					
Effective stick pattern, ft	WP	444.3			
	LP	993.3			
Conditional prob of damage	PCD1	0.814			
Degree of overlap - deflection	No_d	1.955			
Cond prob damage-deflection	PCDd	0.963			
Degree of overlap - range	No_r	2.355			
Prob damage within pattern	Pcds	1.000			
FD_1 CALCULATION					
	RANGE		DEFLECTION		
Sigma (ft)	252.039		222.387		
Effective pattern size LEP/WEP	1000.000		1000.000		
INTEGRAL LIMITS					
s	1000.000		1000.000		
t	0.000		0.000		
a	-2.806		-3.180		
b	0.000		0.000		
Fractional coverage Fr/Fd	0.799		0.823		
FD_1	**0.232**				

Figure 29. Results from Prof. Driels' Excel program. From [6]

Weapon Effectiveness							
Effectiveness Index Type		MAE_b	MAE_f				
Effectiveness Value	**EI**		4666		**# Submunitons causing Damage**	**nx**	17.90349
Length/Width Ratio (Frag Only)	**a**				**Edge Effect Adjustment Factor**	**RA**	74.00591
Effective Target Width, ft	**WET**		420.0118261		**Pd of one submunitions**	**Ps**	0.088631
Effective Target Length, ft	**LET**		328.0118261		**Pd of all submunitions**	**Pcd_wep**	1
Effective Target Area, ft^2	**AET**		137768.8461				
Single Weapon Effective Dim ft	**WB**	434.3095848			**Effective Stick Pattern, ft**	**WP**	444.3096
	LB	389.8365442				**LP**	993.225
	AB	169309.7476				**AP**	441299.4
Conditional Prob of Damage	**PCD 1**	0.81370888			**Degree of Overlap - Deflection**	**Nod**	1.954986
Cond Prob Damage - Deflection	**PCD-d**	0.962568638			**Degree of Overlap - Range**	**Nor**	2.354974
Prob Damage within Pattern	**PCDS**	0.999563477					

EFD Calculation		**Range**	**Deflection**
Sigma (ft)		252.0385471	222.3869533
LEP/WEP		1000	1000
INTEGRAL LIMITS			
s		1000	1000
t		0	0
a		2.805550141	3.179623493
b		0	0
INTEGRALS			
I1		0	0
I2		0.499963707	0.499996549
I3		0.499963707	0.499996549
I4-I5		-0.201020925	-0.1774319
RESULTS			
Fractional Coverage		0.79890649	0.822561197
Combined Fractional Coverage		0.657149479	
EFD		0.231898467	

Figure 30. Results from cluster munitions program.

VIII. PROJECTILES MODULE

A. BACKGROUND

The projectiles module computes the probability of damage from employing guns and rockets against a ground target. This probability of damage is represented as the Probability of Damage (PD). The materials in this chapter are taken from Chapter 14 of [1]. Additional theoretical information can be found in the same source.

B. COMPUTATION OF PROJECTILES EFFECTIVENESS

While the principle to compute the weapons effectiveness of projectiles is the same as that given for previous air-to-ground weapons, there is a difference in the methodology due to the following:

1) The accuracy function is often expressed in the normal plane and expressed in mils. Hence, there is a need to convert the accuracy function from mils to feet.

2) The damage function has to be expressed in the normal plane even though the MAE is expressed in ground plane. Hence, if MAE is being used as the damage function, it has to be converted to the equivalent damage function in the normal plane.

<u>Conversion of Accuracy Function from Mils to Feet</u>

Prior to converting the accuracy function from mils to feet, the overall REP in normal plane is computed to include the ballistic dispersion of the projectiles. This overall REP can then be converted to feet.

$$\mathrm{REP'_N}\ (\mathrm{mils}) = \sqrt{\mathrm{REP}^2 + \sigma_b{}^2} \tag{8.1}$$

$$\mathrm{DEP'_N}\ (\mathrm{mils}) = \sqrt{\mathrm{DEP}^2 + \sigma_b{}^2} \tag{8.2}$$

$$\mathrm{REP'_N}\ (\mathrm{ft}) = \frac{\mathrm{REP'_N}\ (\mathrm{mils}) * \mathrm{SR}}{1000} \tag{8.3}$$

$$\mathrm{DEP'_N}\ (\mathrm{ft}) = \frac{\mathrm{DEP'_N}\ (\mathrm{mils}) * \mathrm{SR}}{1000} \tag{8.4}$$

$$\sigma'_{NR}\ (\text{ft}) = \frac{\text{REP}'_N\ (\text{ft})}{0.6745} \tag{8.5}$$

$$\sigma'_{ND}\ (\text{ft}) = \frac{\text{DEP}'_N\ (\text{ft})}{0.6745} \tag{8.6}$$

Computation of Damage Function

There can be three ways in which the damage function for a projectile can be expressed: MAE (Blast), MAE (Fragmentation) and Vulnerable Area (Av). Both MAE (Blast) and MAE (Fragmentation) are expressed in ground plane and need to be converted to the normal plane as illustrated in Figure 31. This is done by converting the LET on the ground plane to the projected LET' on the normal plane. The WET is the same in both the ground plane and normal plane. The impact angle used for the conversion is also the aircraft dive angle. Av is already in the normal plane and does not need to be converted.

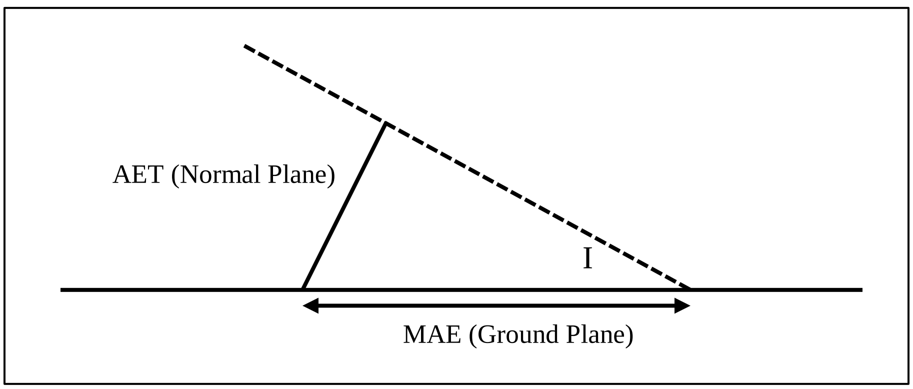


Figure 31. Projection of damage function from ground plane to normal plane.

MAE (Blast):

$$\text{WET} = \text{LET} = \sqrt{\text{MAE}_B} \tag{8.7}$$

$$\text{LET}' = \text{LET} * \sin(\text{Dive Angle}) \tag{8.8}$$

$$\text{AET} = \text{LET}' * \text{WET} \tag{8.9}$$

MAE (Fragmentation):

$$a = \max(1 - 0.8 * \sin \text{dive angle}, 0.3) \tag{8.10}$$

$$\text{LET} = \sqrt{\text{MAE}_\text{F} * \text{a}} \tag{8.11}$$

$$\text{WET} = \frac{\text{LET}}{\text{a}} \tag{8.12}$$

$$\text{LET}' = \text{LET} * \sin(\text{Dive Angle}) \tag{8.13}$$

$$\text{AET} = \text{LET}' * \text{WET} \tag{8.14}$$

Av:

$$\text{WET} = \text{LET} = \sqrt{\text{MAE}_\text{B}} \tag{8.15}$$

$$\text{LET}' = \text{LET} * \sin(\text{Dive Angle}) \tag{8.16}$$

$$\text{AET} = \text{LET}' * \text{WET} \tag{8.17}$$

SSPD Computation

Once the damage function and accuracy function are computed, the SSPD of one round can be computed.

$$\text{SSPD}_\text{X} = 2 * (\text{normcdf}\left(\frac{\text{LET}'}{2}, 0, \sigma'_{\text{NR}}\right) - 0.5) \tag{8.18}$$

$$\text{SSPD}_\text{Y} = 2 * (\text{normcdf}\left(\frac{\text{WET}'}{2}, 0, \sigma'_{\text{ND}}\right) - 0.5) \tag{8.19}$$

$$\text{SSPD}_1 = \text{SSPD}_\text{X} * \text{SSPD}_\text{Y} \tag{8.20}$$

PD Computation for 1 Hit Required on Target

If only one hit is required to damage the target, the probability of damage after firing multiple projectiles can be computed using the survivability rule.

$$\textit{Projectiles Actually Fired} = \text{round}(\text{Rounds Fired} * \text{Weapons Reliability}) \tag{8.21}$$

$$\text{PD} = 1 - (1 - \text{SSPD}_1)^\text{i}, \tag{8.22}$$

$$\textit{where i is the number of projectiles actually fired}$$

PD Computation for >1 Hit Required on Target

If more than one hit is required to damage the target, the probability of damage can be computed using the binomial distribution command.

$$PD_{n\,hit\,kill} = 1 - binocdf(Hits\ Required\ to\ Kill\ Target - 1,\quad Projectiles\ Actually\ Fired\ , SSPD_1) \tag{8.23}$$

C. PROJECTILES MODULE IN MATLAB

The projectiles module is modeled as a function using the equations explained in the previous section. The input commands and the associated outputs for the module are depicted in Table 23. A flowchart of the module is shown in Figure 25.

Input	`Rocket_Projectile_Model(dive_angle, SR, MAE, Damage_Function_Type, Rounds_Fired, Hits_to_Kill_Target, Sigma_b, Weapon_Reliability, REP, DEP)`
Output	`Rocket_Projectile_output = [SSPD_1hit,SSPD_nhit_kill];`

Table 23. Input and output commands for projectiles module.

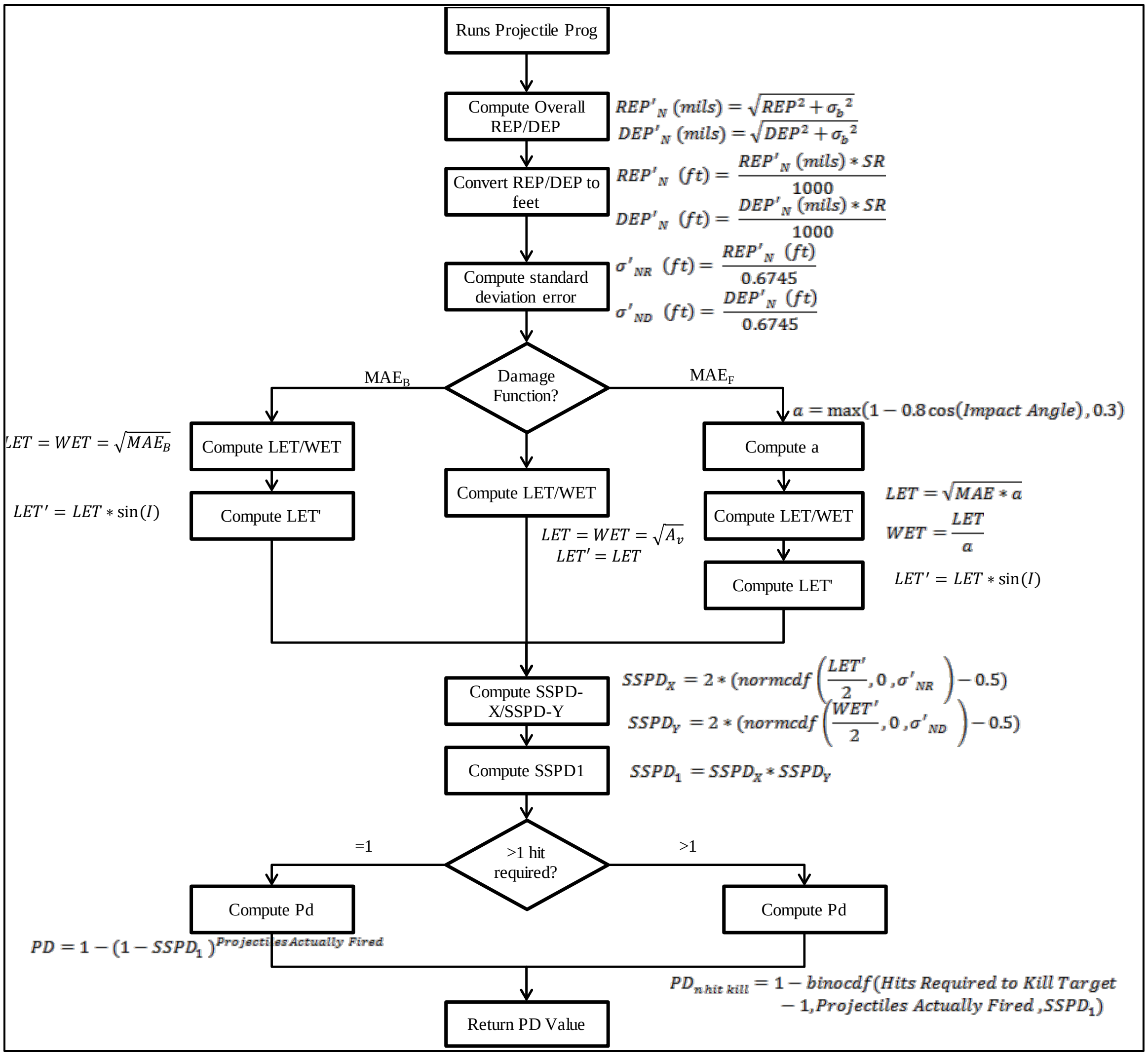


Figure 32. Flow diagram of projectile module.

D. RESULTS OF PROJECTILE MODULE IN MATLAB

The results of the projectile program were compared with an Excel program provided by Prof. Driels [7] and were verified to be good. A print-screen of the result provided by Prof. Driels is illustrated in Figure 33. Figure 26. A print-screen of the result from the projectile program is illustrated in Figure 34.

WEAPON PARAMETERS				
Ballistic dispersion normal, mils	sigma_b	9.000		
Reliability	R	0.830		
AIRCRAFT RELEASE DATA				
Airspeed -KTAS / ft/sec	V	450		759.600
Dive angle, degrees	theta	30		0.524
Number of rounds fired	n	38		
Number of rounds actually fired	nf	32		
Slant range, ft	SR	5000		
TRAJECTORY RESULTS				
Impact angle, degrees	I	30		
Ground range, ft	Gr	4330		
Altitude, ft	y_bar	2500		
DELIVERY ACCURACY				
REP/DEP normal, mils (no BD)		12.00	9.00	
CEP, normal, mils (no BD)		0.00	0.00	0.00
REP/DEP, normal, mils (with BD)		13.448	10.856	
WEAPON EFFECTIVENESS				
Effectiveness type		MAE (blast or frag)	Av_n	
Effectiveness value - enter one only		500.00	0.000	
Effectiveness in the normal plane		250.000	0.000	
Number of hits to kill the target (>0)	n_rk	1		
Target area in normal plane	AET	250.000	0.000	
SINGLE SORTIE EFFECTIVENESS				
		Range	Deflection	
Target dimension, ft	T_d	15.811	15.811	
Sigma at the slant range, ft	sigma_sl	99.689	80.474	
PD_1		0.063	0.078	
Prob. exactly 1 round hits target	PD_1	0.00495		
Prob. At least 1 of nf hits target	P_{HIT}	0.142		
Prob. Target killed	PD	0.142		

Figure 33. Results from Prof. Driels' Excel program. From [7]

Trajectory Results				
Impact Angle	deg	30		
Ground Range		2500		
Altitude		4330.127		
Slant Range				
		-	-	-
Additional Results		-	-	-
Number of Rounds Fired		38	-	-
Number of Rounds Actually Fired		32	-	-
		-	-	-
		-	-	-
		-	-	-
		-	-	-
		-	-	-
		-	-	-
		-	-	-
		-	-	-
		-	-	-
		-	-	-

Delivery Accuracy		
CEP, normal, mils (No BD)		
DEP, normal, mils (No BD)		9
REP, normal, mils (No BD)		12
DEP	mils	54.27959
REP	ft	67.24042

Weapon Effectiveness			
Effectiveness Index Type		MAE(Blast)	
Effectiveness Value		500	
Effectiveness in the normal plane		250	
Target Area in the Normal Plane			

SSPD Calculation		Range	Deflection
Target Dimensions, LET/WET		11.18033989	22.36067977
Sigma at Slant Range		99.68928287	80.47382264
SSPD-x/SSPD-y		0.044718687	0.110495633
Prob exactly 1 rnd hits target		0.00494122	
Prob at least 1 rnd hits target		0.146586397	
Prob Target Killed		0.146586397	

Figure 34. Results from projectile program.

IX. INDIRECT FIRE WEAPONS PROGRAM

A. BACKGROUND

The indirect fire weapons program computes the probability of damage from employing surface-to-surface (SS) weapons such as mortars and artillery. Such weapons are typically employed as a battery and not as an individual weapon. The indirect fire weapons can be deployed against an area target or a point target, both of which use different methodologies to compute the weapon effectiveness. The methodology for area targets is modeled after the existing Superquickie2 program. The methodology for point targets uses the Monte Carlo simulation program. The materials in this chapter are taken from Chapter 17 of [1]. Additional theoretical information can be found in the same source.

B. INDIRECT FIRE WEAPONS AGAINST AREA TARGET

The indirect fire weapons program against an area target is modeled after the Superquickie2 program, which is similar to the stick delivery program. The type of warheads used can be a unitary warhead or an improved conventional munitions (ICM) warhead; the latter is somewhat akin to cluster munitions for AS weapons. The principle to compute the weapons effectiveness for SS remains the same as for AS weapons which involves computing the damage function and accuracy function of the weapon, as illustrated in the following subsection. The weapon effectiveness for indirect fire SS weapons against area target is expressed as Expected Fractional Damage (EFD).

1. Weapon Effectiveness of Unitary Warhead

<u>Computing Total MPI Error in Range and Deflection</u>

The total MPI error for range and deflection can be calculated as follows:

$$Total\ MPI\ Error\ (Range) = \sqrt{{MPI\ Error_{Range}}^2 + 0.328 * TLE^2} \qquad (9.1)$$

$$Total\ MPI\ Error\ (Deflection) = \sqrt{{MPI\ Error_{Deflection}}^{2} + 0.328 * TLE^{2}} \tag{9.2}$$

The TLE term is normally provided as a CEP. Hence, the term 0.328 is required to convert the TLE into CEP due to its equivalent TLE in REP and DEP.

There is a difference between MPI error and precision error. The MPI error is an occasion-to-occasion error where it is constant for a particular occasion such as a volley of six rounds. The variability of each of these six rounds is the precision error of the weapon which is not constant for that occasion. Hence, for one occasion of a volley consisting of six rounds, each round have the same MPI error but each round has a different precision error. More than one volley can be fired on one occasion.

<u>Computing LET and WET</u>

Similar to AS weapons, the unitary warhead consists of both fragmentation and blast kill mode. However, there are some subtle differences in the computation of the LET and WET for a fragmentation warhead, particularly in the aspect ratio computation.

Blast Warhead:

$$LET = \sqrt{AL} \tag{9.3}$$

$$WET = LET \tag{9.4}$$

Fragmentation Warhead:

$$a = 1 - 0.8cos\theta \tag{9.5}$$

$$LET = \sqrt{AL * a} \tag{9.6}$$

$$WET = LET/a \tag{9.7}$$

Note that the computation of the aspect ratio a does not require a comparison between the computed value and 0.3 as SS weapons do not have an impact angle physical limitation as compared to AS weapons.

<u>Computing L_b, W_b and PCD1</u>

$$L_b = LET + k * REP\_p \tag{9.8}$$

$$W_b = WET + k * DEP\_p \tag{9.9}$$

The k term is a pattern adjustment factor that is a user input that can be obtained from tabular data.

$$PCD1 = \frac{LET * WET}{L_b * W_b} \tag{9.10}$$

<u>Computing L_v and W_v</u>

The volley length and width computation is dependent on the dimension type of the target. The two types of dimensions available in the program are rectangular and circular.

Rectangular Target Dimension:

$$L_v = (Nwr - 1) * \frac{L_A}{Nwr} \tag{9.11}$$

$$W_v = (Nwd - 1) * \frac{W_A}{Nwd} \tag{9.12}$$

Circular Target Dimension:

$$L_v = R_A \tag{9.13}$$

$$W_v = R_A * cos(90 - \frac{360}{n_r}) \tag{9.14}$$

<u>Computing L_{vp} and W_{vp}</u>

$$L_{vp} = L_v + L_b \tag{9.15}$$

$$W_{vp} = W_v + W_b \tag{9.16}$$

<u>Computing n_{od} and $P_{CD/d}$</u>

$$n_{od} = N_{wd} * W_b / W_{vp} \tag{9.17}$$

For $n_{od} < 1$:

$$P_{CD/d} = N_{wd} P_{CD1} \frac{W_b}{W_{vp}} \tag{9.18}$$

For $n_{od} \geq 1$:

$$P_{CD/d} = 1 - (1 - P_{CD1})^{n_{od}} \tag{9.19}$$

Computing n_{or} and P_{CDS}

$$n_{or} = N_{wr} * L_b / L_{vp} \tag{9.20}$$

For $n_{or} < 1$:

$$P_{CDS} = n_r P_{CD/d} \frac{L_b}{L_{vp}} \tag{9.21}$$

For $n_{or} \geq 1$:

$$P_{CDS} = 1 - \left(1 - P_{CD/d}\right)^{n_{or}} \tag{9.22}$$

Expected Fractional Coverage (EFC) Computation

The methodology to compute EFC is the same as the one used for the single weapon area target module and will not be detailed here. One factor to note is that the computation of LEP and WEP is slightly different as shown below.

$$Enlarged\ Weapon\ Lethal\ Length,\ LEP = \max(L_{VP}, LA) \tag{9.23}$$

$$Enlarged\ Weapon\ Lethal\ Width,\ WEP = \max(W_{VP}, WA) \tag{9.24}$$

Expected Fractional Damage (EFD) Computation

The final EFD computation is similar to that used for the single release of cluster weapon as follows, with n_v being the number of volleys fired:

$$\mathrm{EFD} = \mathrm{EFC} * (1 - (1 - \mathrm{R} * \mathrm{P_{CDS}})^{\mathrm{n_v}} \tag{9.25}$$

The main difference in the computation of EFD for indirect fire weapons is that the volleys being fired are considered to be dependent events until the gun or launcher is reloaded. In contrast, subsequent deliveries of weapons for AS deliveries are considered as independent events as the aircraft cannot be at the exact same weapon release location.

2. Weapon Effectiveness of ICM Warhead

The effectiveness computation of ICM warheads is generally the same as the computation for a unitary warhead with the exception that the computation of the conditional damage probability (PCD1) is different as you need to account for all the submunitions. This involves computing the probability of damage of one submunition before computing the damage within the pattern of all the submunitions. This overall probability is then adjusted to account for the effects of the precision error on the lethal area to get PCD1.

Computing PCD1

$$P_S = \frac{R_S * A_L}{L_{SP} * W_{SP}} \tag{9.26}$$

$$P_{DP} = 1 - \exp(-N_S P_S) \tag{9.27}$$

$$L_b = L_{SP} + k * REP\_p \tag{9.28}$$

$$W_b = W_{SP} + k * DEP\_p \tag{9.29}$$

$$PCD1 = P_{DP} * \frac{L_{SP} * W_{SP}}{L_b * W_b} \tag{9.30}$$

Once PCD1 is computed, the subsequent steps to compute the EFD are the same as that for a unitary warhead.

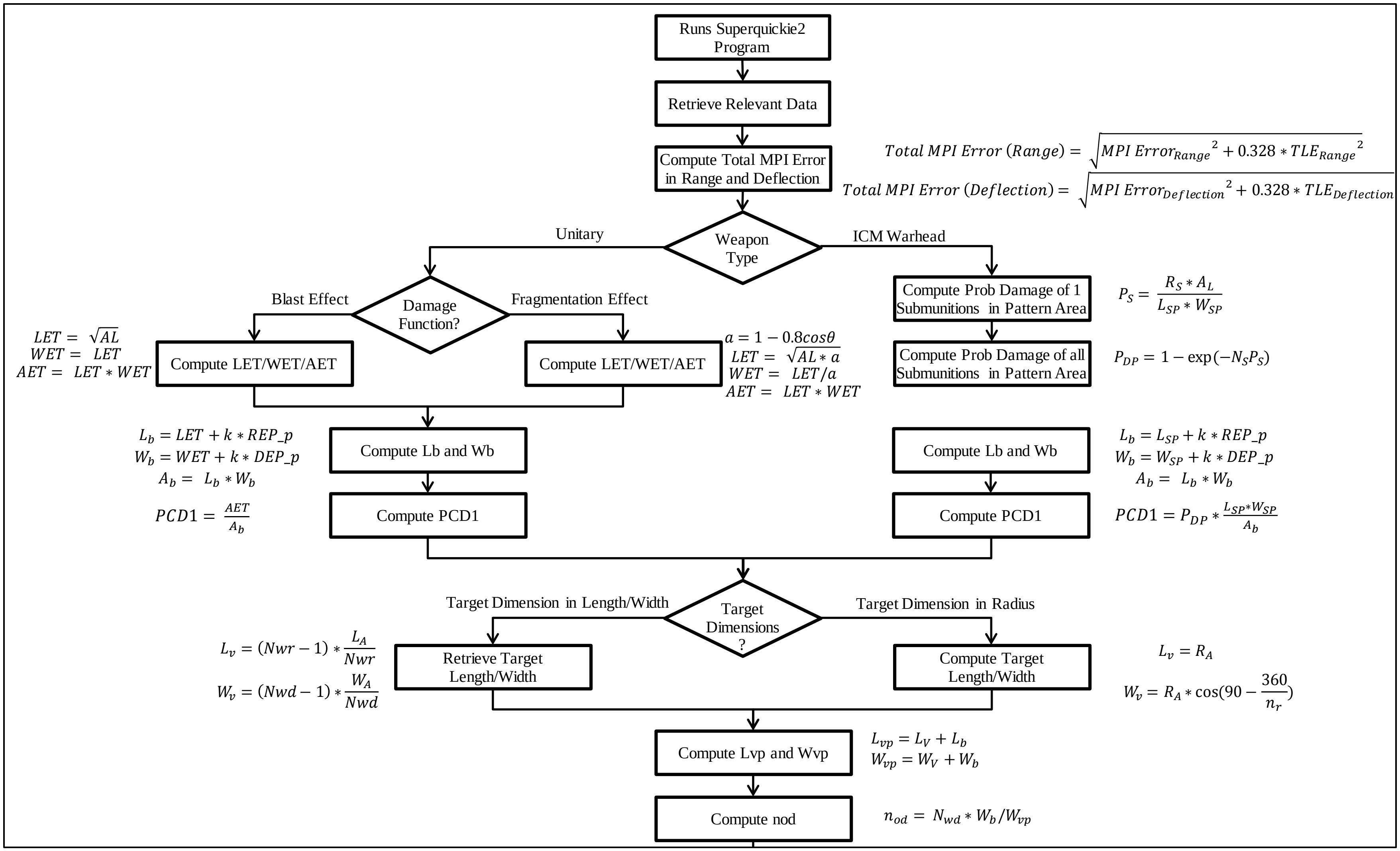
3. Indirect Fire Weapons against Area Targets Module in MATLAB

The indirect fire weapons module is modeled as a function using the equations explained in the previous section. The input commands and the associated outputs for the module are depicted in Table 24. A flowchart of the module is shown in Figure 35.

Input	`Indirect_Weapons_Model(Initial_Data, Accuracy_Data, Rounds_Data)`
Output	`Indirect_Weapons_output = [Weapon_Pattern_Output, Overlap_Data_Output, Integral_Range_Output, EF_R, EF_D, EFC, FD, P_s, P_DP];`

Table 24. Input and output commands for indirect fire weapon against area target module.

Note that the inputs Initial_Data, Accuracy_Data and Rounds_Data are all arrays. The details of the inputs that form these arrays can be found in the appendices.



(Continued on next page)

(Continued from previous page)

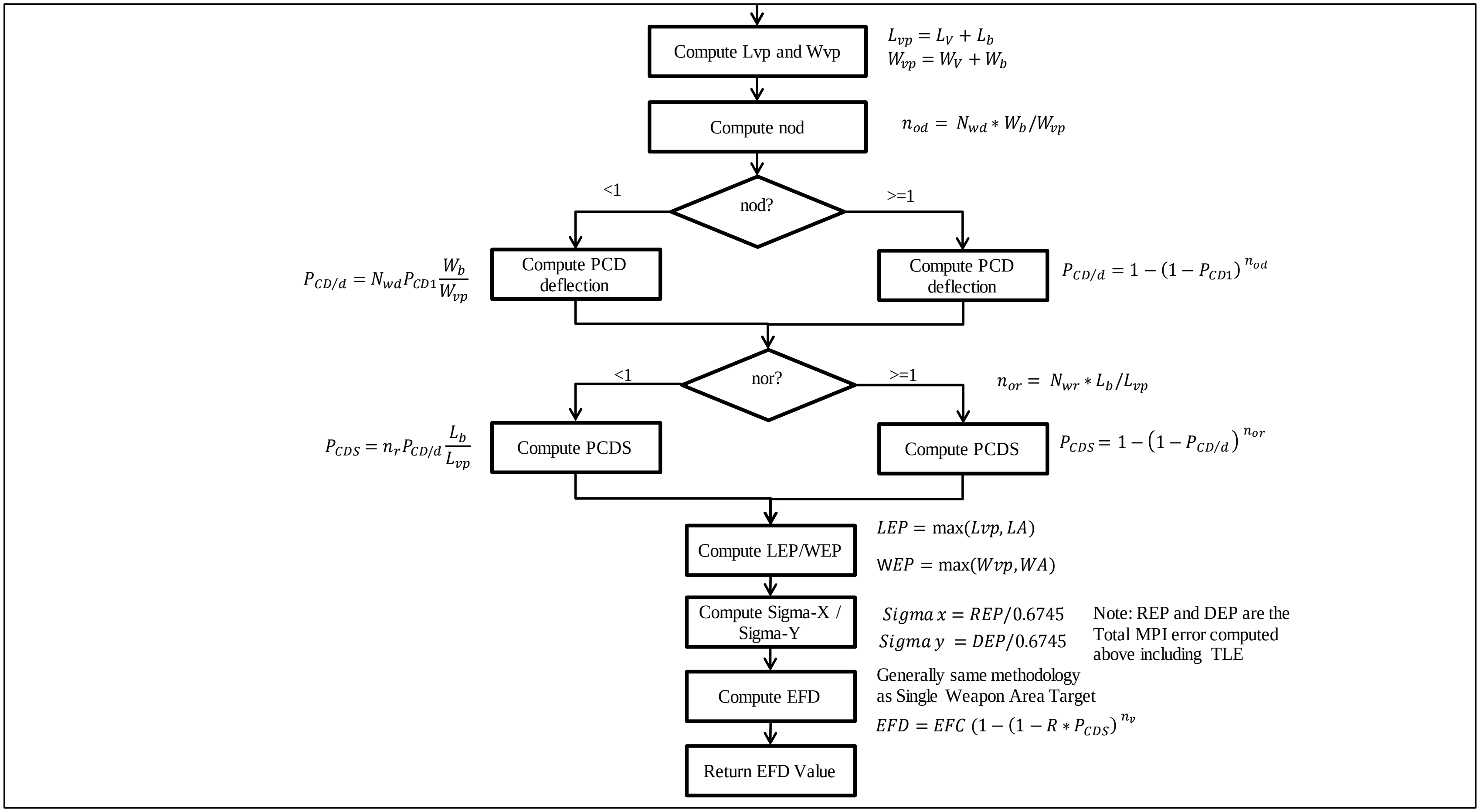


Figure 35. Flow diagram of indirect fire weapon against area target module.

4. Results of Indirect Fire Weapons against Area Targets Module

The results of the indirect fire weapons against area targets module were compared with an Excel program provided by Prof. Driels [8] and were verified to be good. A print-screen of the result provided by Prof. Driels is illustrated in Figure 36. Figure 26. A print-screen of the result from the indirect fire weapons against area targets program is illustrated in Figure 37.

# rounds/volley	nr	6
# of volleys fired	nv	9
# aimpoints in range direction	Nwr	2
# aimpoints in deflection direction	Nwd	3
Target radius (if specified)		100
Target length used	LT	177.2
Target width used	WT	177.2
Range from weapon to target (km)	Range	16
TLE	TLE	0
Precision error range	REP_p	45.2
Precision error deflection	DEP_p	7.4
MPI error range	REP_m	78.6
MPI error deflection	DEP_m	45.7
Total MPI error range (including TLE)	REP_tm	78.600
Total MPI error deflection (including TLE)	DEP_tm	45.700
Round reliability	Rr	0.85
Submunition reliability	Rs	0.9
Number of submunitions/round	Ns	60
Single round pattern length	Lsp	182.52
Single round pattern width	Wsp	106.32
Single round submunition lethal area	AL	44
Pd due to one submumition in pattern	Ps	0.002
Pd due to all submunitions	Pdp	0.115
Volley length	LV	100.00
Volley width	WV	86.60
Pattern adjustment factor	k	3.000
Single round adjusted pattern length	Lb	318.120
Single round adjusted pattern width	Wb	128.520
Conditional probability of damage	PCD1	0.055
Volley damage pattern length	Lvp	418.12
Volley damage pattern width	Wvp	215.12
Sigma range	116.531	67.754
LEP/WEP	418.120	215.124
Integral limits		
s	297.683	196.185
t	120.437	18.940
a	1.806	2.047
b	0.731	0.198
Integrals		
integral #1	-1.648	-0.267
integral #2	1.418	0.673
integral #3	1.418	0.673
integral #4-integral#5	-0.287	-0.289
Fractional coverage	**0.899**	**0.790**
Overlap in deflection	no_d	1.792
Prob damage in deflection	PCDd	0.096
Overlap in range	no_r	1.522
Pattern probability of damage	PCD	0.142
Probability of damage for nv volleys		**0.686**
Expected fractional damage or casualties	Fd, Fc	**0.488**

Figure 36. Results from Prof. Driels' Excel program. From [8]

Basic Data			
# rounds/volley	nr		6
# volleys fired	nv		9
# aim points in range direction	Nwr		2
# aim points in deflection direction	Nwd		3
Target Length	LT	ft	
Target Width	WT	ft	
Target Radius	RT	ft	100
Accuracy Computation Data			
Precision Error Range	REP_p	ft	45.2
Precision Error Deflection	DEP_p	ft	7.4
MPI Error Range	REP_m	ft	78.6
MPI Error Deflection	DEP_m	ft	45.7
TLE	TLE	ft	0
Total MPI Error Range (Including TLI)	REP_tm	ft	78.6
Total MPI Error Deflection (Including TLI)	DEP_tm	ft	45.7
Rounds Parameters			
Round Reliability	Rr		0.85
Submunitions Reliability	Rs		0.9
# of Submunitions/Round	Ns	ft^2	60
Single Round Pattern Length	Lsp	ft	182.52
Single Round Pattern Width	Wsp	ft	106.32
Single Round Submunitions Lethal Area	AL		44

Weapon Effectiveness								
Volley Length	**LV**	**ft**	100		**Volley Width**	**WV**	**ft**	86.6025404
Lethal Area Aspect Ratio	**a**							
Single Round Adjusted Pattern Length	**Lb**	**ft**	318.12		**Single Round Adjusted Pattern Width**	**Wb**	**ft**	128.52
Single Round Adjusted Pattern Area	**Ab**	**ft^2**	40884.7824		**Conditional Probability of Damage**	**PCD1**		0.05469764
Pd for 1 submunitions in Pattern	**Ps**		0.002040656		**Pd for all submunitions in Pattern**	**Pdp**		0.11524043
Volley Damage Pattern Length	**LVP**	**ft**	418.12		**Volley Damage Pattern Width**	**WVP**	**ft**	215.12254
Overlap in Deflection	**nod**		1.792280806		**Prob Damage in Deflection**	**PCD/d**		0.09590117
Overlap in Range	**nor**		1.521668421		**Pattern Prob of Damage**	**PCDS**		0.14222156
Effective Weapon Length	**LEP**	**ft**	418.12		**Effective Weapon Width**	**WEP**	**ft**	215.12254

		Range	Deflection
Sigma		116.53	67.75389177
Integeral Limits			
s		297.68	196.1839627
t		120.44	18.93857764
a		1.8063	2.047454497
b		0.7308	0.19765059
Integrals			
Integral #1		0.6986	0.22015432
Integral #2		0.2441	0.42949113
Integral #3		0.2441	0.42949113
Integral #4-Integral#5		-0.287	-0.2887043
Expected Fractional Coverage		0.8995	0.790432276
Expected Fractional Coverage		0.711	
Fractional Damage		0.488	

Figure 37. Results from indirect fire weapon vs. area target program.

C. INDIRECT FIRE WEAPONS AGAINST SINGLE POINT TARGET

The module for an indirect fire weapon against a single point target is modeled using Monte Carlo simulation. The type of warhead used can be either a fragmentation unitary warhead or blast unitary warhead. The weapon effectiveness of indirect fire weapons against a single point target is expressed as probability of damage (PD).

Prior to the Monte Carlo simulation, the standard deviation for the MPI error and precision error defining the accuracy of the weapon and the LET and WET defining the damage capability of weapon must be computed.

<u>Defining Accuracy of Weapon</u>

$$\text{Total MPI Error (Range)} = \sqrt{\text{MPI Error}_{\text{Range}}{}^2 + 0.328 * \text{TLE}_{\text{Range}}{}^2} \quad (9.31)$$

$$\text{Total MPI Error (Deflection)} = \sqrt{\text{MPI Error}_{\text{Deflection}}{}^2 + 0.328 * \text{TLE}_{\text{Deflection}}{}^2} \quad (9.32)$$

$$\sigma_{\text{MPI}-\text{x}} = \frac{\text{Total MPI Error (Range)}}{0.6745} \quad (9.33)$$

$$\sigma_{\text{MPI}-\text{y}} = \frac{\text{Total MPI Error (Deflection)}}{0.6745} \quad (9.34)$$

$$\sigma_{\text{PE}-\text{x}} = \frac{\text{Total Precision Error (Range)}}{0.6745} \quad (9.35)$$

$$\sigma_{\text{PE}-\text{y}} = \frac{\text{Total Precision Error (Deflection)}}{0.6745} \quad (9.36)$$

<u>Defining Damage Capability of Weapon</u>

Blast Warhead:

$$\text{LET} = \sqrt{\text{AL}} \quad (9.37)$$

$$\text{WET} = \text{LET} \quad (9.38)$$

Fragmentation Warhead:

$$\text{a} = 1 - 0.8\cos\theta \quad (9.39)$$

$$\text{LET} = \sqrt{\text{AL} * \text{a}} \quad (9.40)$$

$$\mathrm{WET} = \mathrm{LET}/\mathrm{a} \tag{9.41}$$

Commence Monte Carlo Simulation

The Monte Carlo simulation will be run based on the number of iterations that is input by the user. When the number of iterations increases, the computed PD will be more accurate at the expense of simulation time. An iteration can be considered as an occasion when the weapons have completed firing all the rounds specified by the user.

For each iteration, the MPI error is constant. Hence, a mean point of impact is generated for each iteration based on a normal distribution using the defined MPI error standard deviation as shown.

$$\mathrm{x_{random}} = \mathrm{random}('\mathrm{Normal}',\ 0,\ \sigma_{\mathrm{MPI-x}}) \tag{9.42}$$

$$\mathrm{y_{random}} = \mathrm{random}('\mathrm{Normal}',\ 0,\ \sigma_{\mathrm{MPI-y}}) \tag{9.43}$$

This generated point will be the same MPI for all the rounds fired in that particular iteration. Once all the rounds for the iteration are fired, another random MPI will be generated until all the iterations are completed.

Once the MPI is defined, the exact impact point for each of the rounds fired for that iteration is computed as follows:

$$x_{precision-random} = random('Normal',\ x_{random},\ \sigma_{PE-x}) \tag{9.44}$$

$$y_{precision-random} = random\left('Normal',\ y_{random},\ \sigma_{PE-y}\right) \tag{9.45}$$

Note that each of these impact points can be different for one particular iteration as the precision for each weapon fired is not constant for a particular iteration, unlike the MPI.

Each impact point is then compared with the defined LET and WET. If the impact point lies within the area defined by the LET and WET, the hit counter will increase by one and the particular iteration ends.

$$and(abs(x_{precision-random}) \le \frac{LET}{2}, abs(y_{precision-random}) \le \frac{WET}{2})$$

$$PD = PD + 1$$

Break

When all the rounds for the particular iteration have been fired, the next iteration will start and the cycle will continue until all the iterations are completed.

Computing Weapon Effectiveness

Once all the iterations are completed, the weapon effectiveness is computed by averaging the total number of hits with the total number of iterations.

$$Average\ PD = \frac{Total\ PD}{Total\ Iterations} \tag{9.46}$$

The methodology for the Monte Carlo program is illustrated in the flow diagram for the programming of the indirect fire weapon module against a point target in Figure 38.

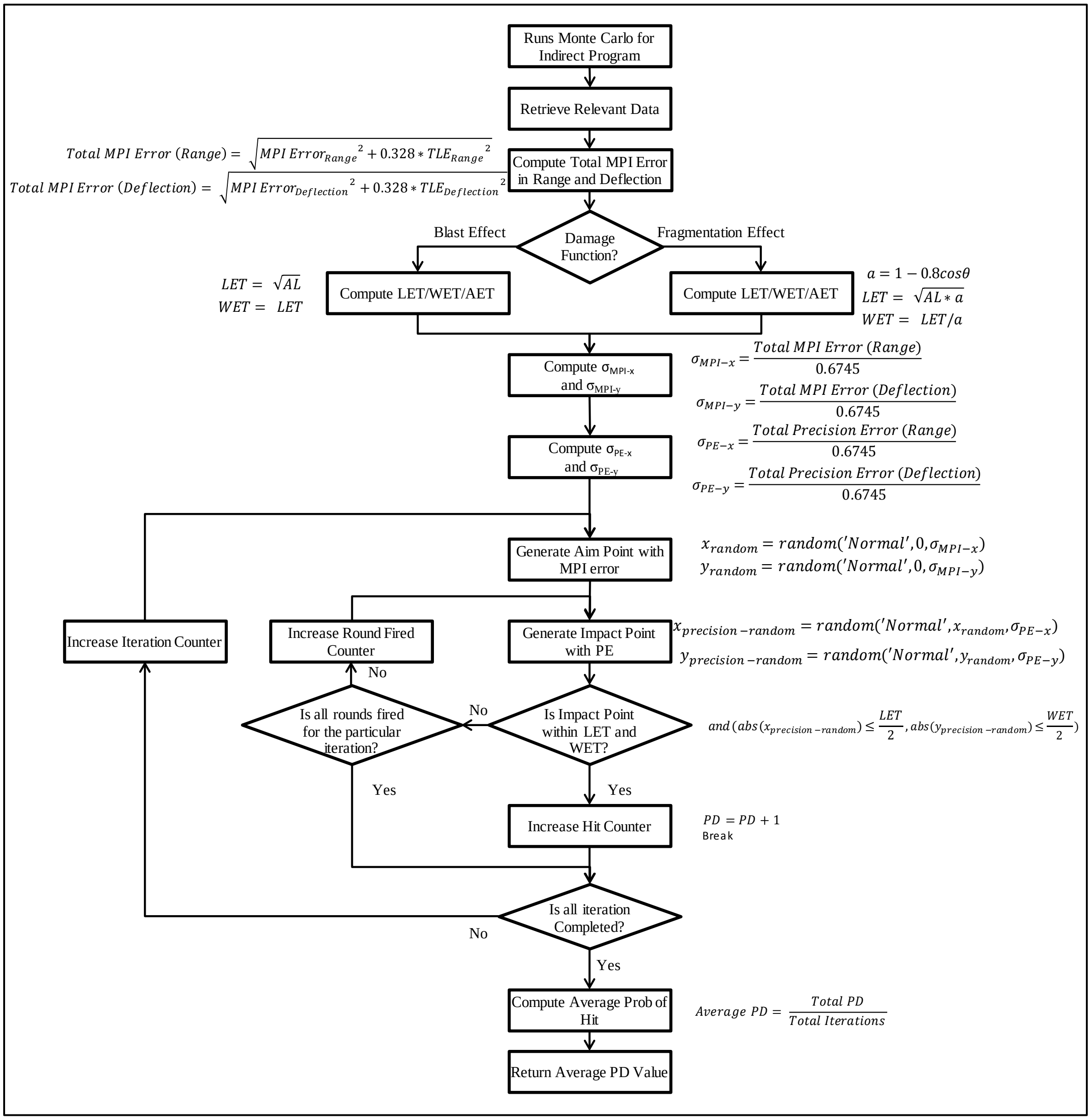


Figure 38. Flow diagram of indirect fire weapon against point target module.

1. Indirect Fire Weapons against Point Targets Module in MATLAB

The indirect fire weapons module is modeled as a function using the methodology explained above. The input commands and the associated outputs for the module are depicted in Table 24.

Input	`Indirect_Monte_Carlo_Model(Initial_Data, Accuracy_Data, Rounds_Data)`
Output	`Indirect_Monte_Carlo_output = Average_P_D;`

Table 25. Input and output commands for indirect fire weapon against point target module.

Note that the inputs Initial_Data, Accuracy_Data and Rounds_Data are all arrays. The details of the inputs that form these arrays can be found in the Appendices.

2. Results of Indirect Fire Weapons against Point Targets Module

There is no existing equivalent program to verify the results of the module. Hence, a baseline case study was developed to analyze the results when some parameters are changed to see if they are reasonable, with the results tabulated in Table 26.

Parameters	**Case Study**							
	1 (Baseline)	**2**	**3**	**4**	**5**	**6**	**7**	**8**
#Rounds / Volley	4	4	4	4	4	4	2	4
Volleys Fired	2	2	2	2	2	4	2	2
Precision Error (Range) (ft)	10	5	15	10	10	10	10	10
Precision Error (Deflection) (ft)	10	5	15	10	10	10	10	10
MPI Error (Range) (ft)	5	2	10	5	5	5	5	5
MPI Error (Deflection) (ft)	5	2	10	5	5	5	5	5
TLE (ft)	0	0	0	0	0	0	0	0
Lethal Area (ft^2)	50	50	50	25	100	50	50	50
Number of Iterations	10000	10000	10000	10000	10000	10000	10000	100000
Probability of Damage (PD)	0.207	0.623	0.085	0.104	0.373	0.370	0.109	0.208

Table 26. Results of indirect fire weapon against point target module.

The results seem reasonable based on the following observations:

1) When the MPI error and Precision error are reduced in Case Study 2 and increased in Case Study 3, the PD increases and decreases, respectively.

2) When the lethal area is reduced in Case Study 4 and increased in Case Study 5, the PD decreases and increases, respectively.

3) When the total number of shots fired is increased in Case Study 6 and reduced in Case Study 7, the PD increases and decreases, respectively.

4) When the number of iterations is increased to 100,000 from 10,000, the computed PD remains relatively constant, increasing from 0.207 to 0.208. Hence, the number of iterations (Qty: 10000) used to test the results should be representative of the behavior of the Monte Carlo program.

THIS PAGE INTENTIONALLY LEFT BLANK

X. DIRECT FIRE WEAPON PROGRAM

A. BACKGROUND

The direct fire weapons program computes the probability of damage from employing surface-to-surface (SS) weapons. Specifically, it evaluates the effectiveness of burst firing weapons against infantry troops using the FBAR methodology, which employs a Monte Carlo simulation. The methodology for point targets uses a Monte Carlo simulation program. The materials in this chapter are taken from Chapter 18 of [1]. Additional theoretical information can be found in the same source.

B. COMPUTATION OF DIRECT FIRE USING MONTE CARLO

The direct fire program is modeled after the FBAR program, which uses a Monte Carlo program. In each iteration, the program assigns the target location and the aim point in a uniformly random manner as illustrated in Figure 39. It then assesses whether the aim point hits the target on the ground plane. If the aim point hits the target, it assesses whether the target is incapacitated using a random number generator and the target's probability of incapacitation given a hit. Whenever the target is hit, it is taken out of the particular iteration. The total number of targets being incapacitated is then averaged over the total number of iterations to provide the probability of incapacitation. The type of rounds used can be the typical impact rounds or fragmentation rounds. The entire methodology for the direct fire program is illustrated in Figure 40.

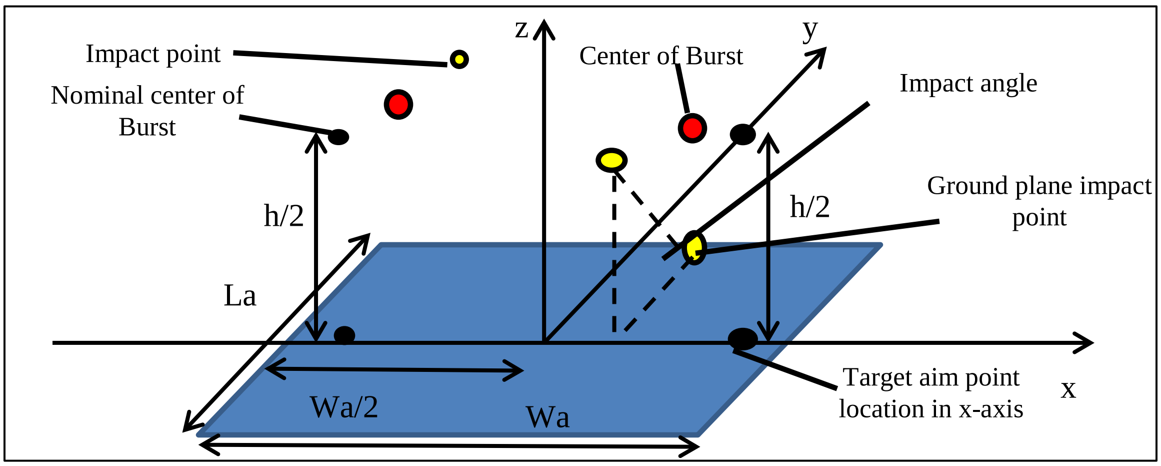
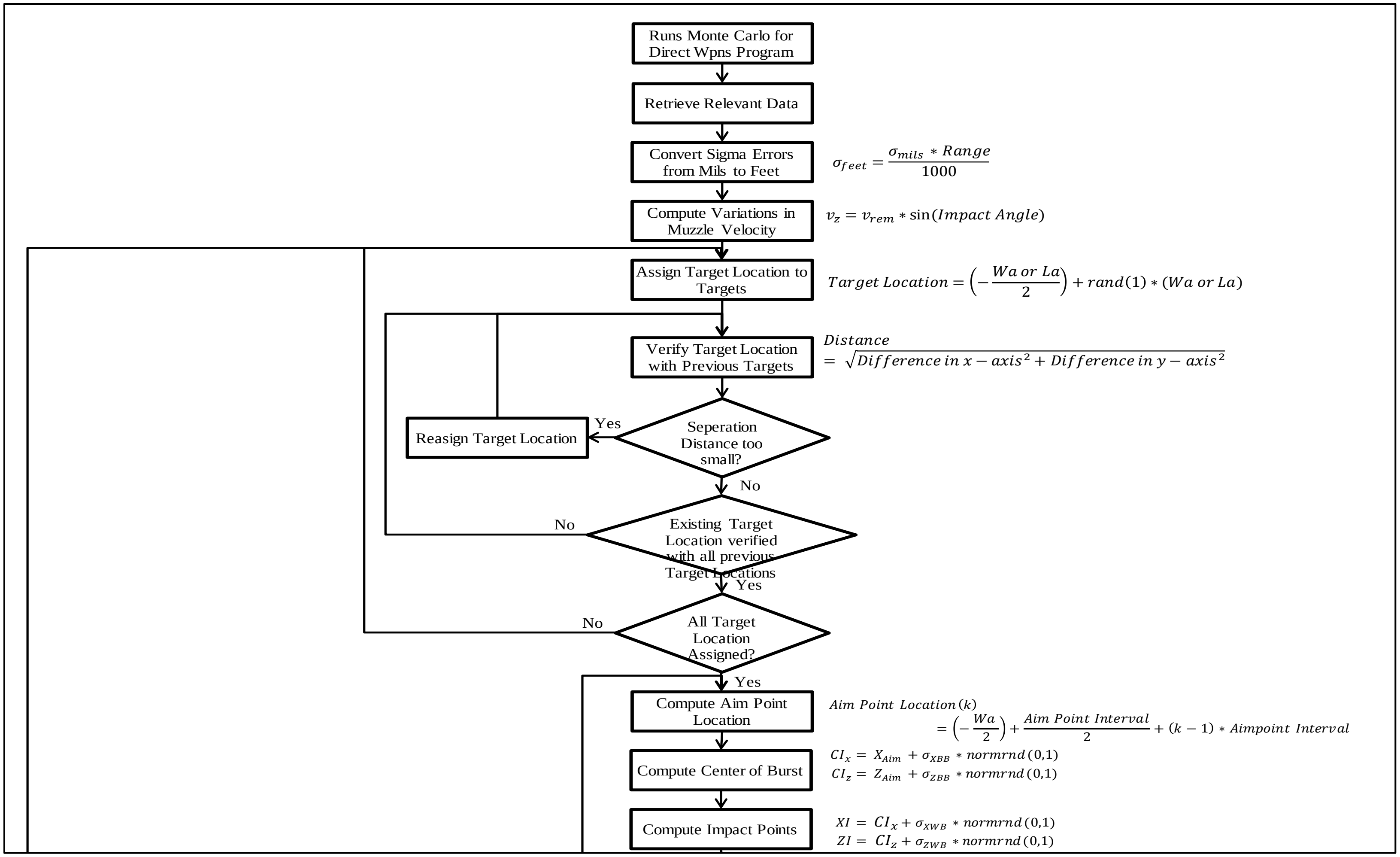
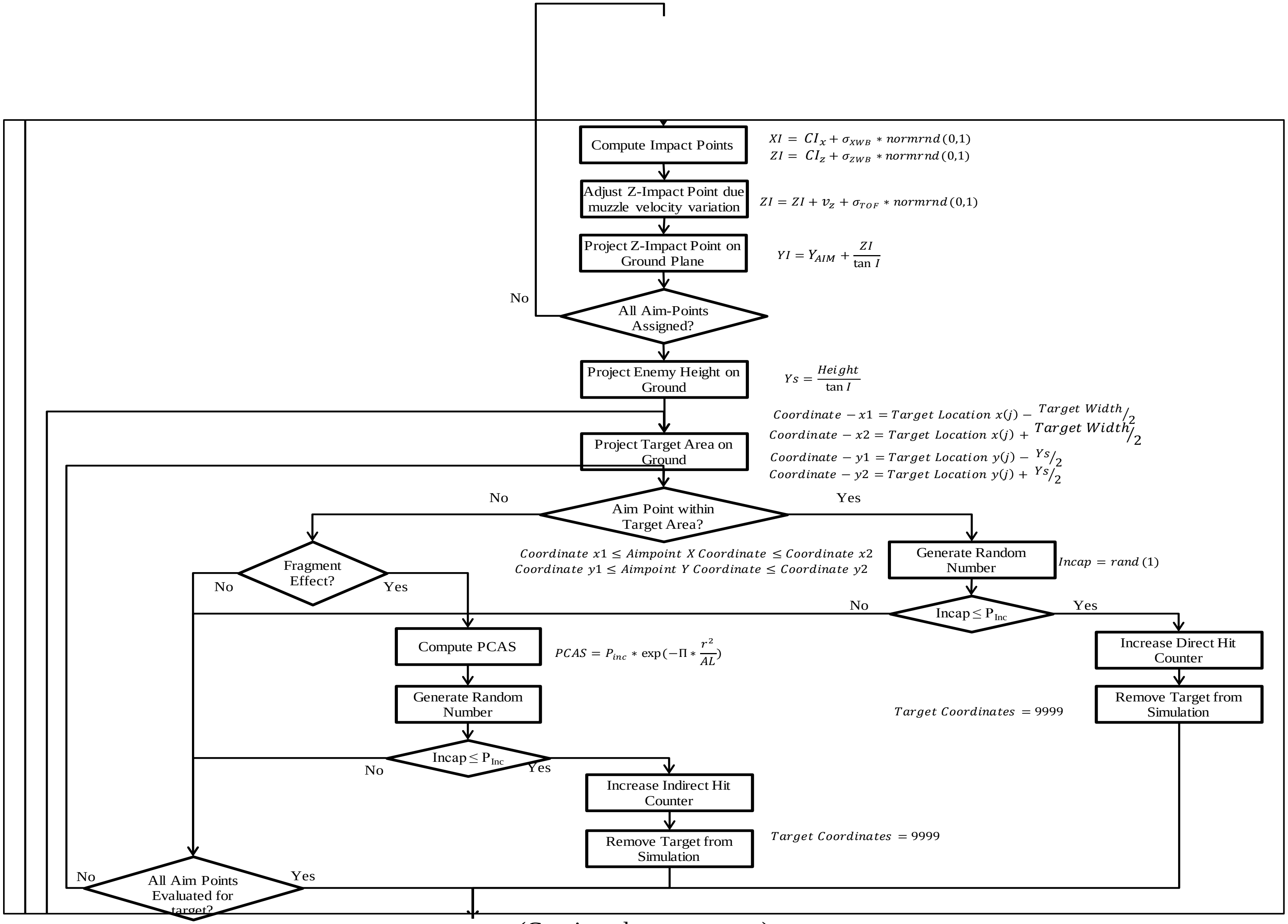


Figure 39. Flow diagram of direct fire weapons module.



(Continued on next page)
(Continued from previous page)



(Continued on next page)
(Continued from previous page)

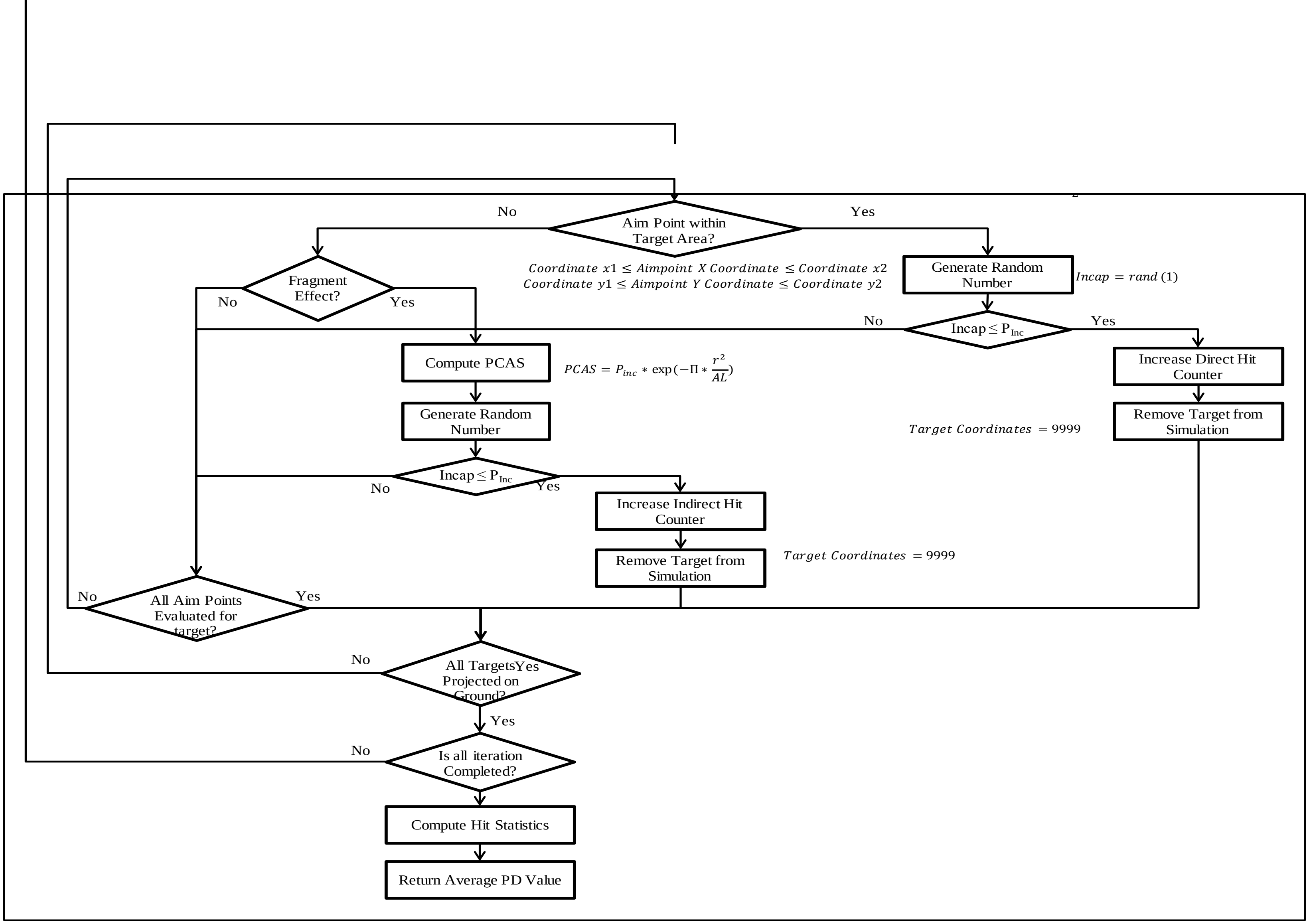


Figure 40. Flow diagram of direct fire weapons module.

Placement of Targets

At the start of each iteration, the targets are placed in a uniformly random manner.

$$\text{Target Location} - \text{x} = \left(-\frac{\text{Wa}}{2}\right) + \text{rand}(1) * (\text{Wa}\ ) \tag{10.1}$$

$$\text{Target Location} - \text{y} = \left(-\frac{\text{La}}{2}\right) + \text{rand}(1) * (\ \text{La}) \tag{10.2}$$

Each target location is then assessed to ensure it is not too close to a previously assigned target location. If it is too close, the target location is re-assigned.

$$Distance = \sqrt{(Difference\ in\ target\ location\ \ x - axis)^2 + (Difference\ in\ target\ location\ y - axis)^2} \tag{10.3}$$

Placement of Aim Points and Centre of Bursts

The aim points are uniformly distributed along the x-axis illustrated in Figure 39. The nominal center of bursts are located at h/2 above the aim points where h is the height of the target.

$$\text{Aim Point}_\text{x} = \left(-\frac{\text{Wa}}{2}\right) + \frac{\text{Aim Point Interval}}{2} + ((\text{k} - 1) * \text{Aim Point Interval}) \tag{10.4}$$

where

$$\text{Aim Point Interval} = \frac{\text{Wa}}{\text{Number of Aim Points}} \tag{10.5}$$

The actual center of burst location (CI) is then computed taking into the account the standard deviation in the MPI error. Note that the standard deviation errors are normally distributed and not uniformly distributed. There is also no need to compute y axis as the impact points are all located at a point where y = 0.

$$\text{CI}_\text{x} = \text{Aim Point}_\text{x} + \sigma_{\text{MPI}-\text{x}} * \text{normrnd}(0,1) \tag{10.6}$$

$$\text{CI}_\text{z} = \text{Aim Point}_\text{z} + \sigma_{\text{MPI}-\text{z}} * \text{normrnd}(0,1) \tag{10.7}$$

Actual Impact Points and Projection on Ground Plane

The actual impact points are then computed taking into account the standard deviation of the precision error. Again, the standard deviation for the precision error is normally distributed.

$$XI = CI_x + \sigma_{PE-x} * \mathrm{normrnd}(0,1) \tag{10.8}$$

$$ZI = CI_z + \sigma_{PE-z} * \mathrm{normrnd}(0,1) \tag{10.9}$$

The z-position of the impact point has to be adjusted to account for the effect of the variation in the muzzle velocity, which can be computed as follows:

$$v_z = v_{rem} * \sin \text{Impact Angle} \tag{10.10}$$

$$\text{Adjusted ZI} = ZI * v_z * \sigma_{TOF} * \mathrm{normrnd}(0,1) \tag{10.11}$$

The impact point can then be projected on the ground plane with reference to the impact angle.

$$YI = \frac{\text{Adjusted ZI}}{\tan(\text{Impact Angle})} \tag{10.12}$$

Projection of Target on Ground Plane

The target has to be projected on the ground to evaluate whether the impact point has hit the target as illustrated in Figure 41.

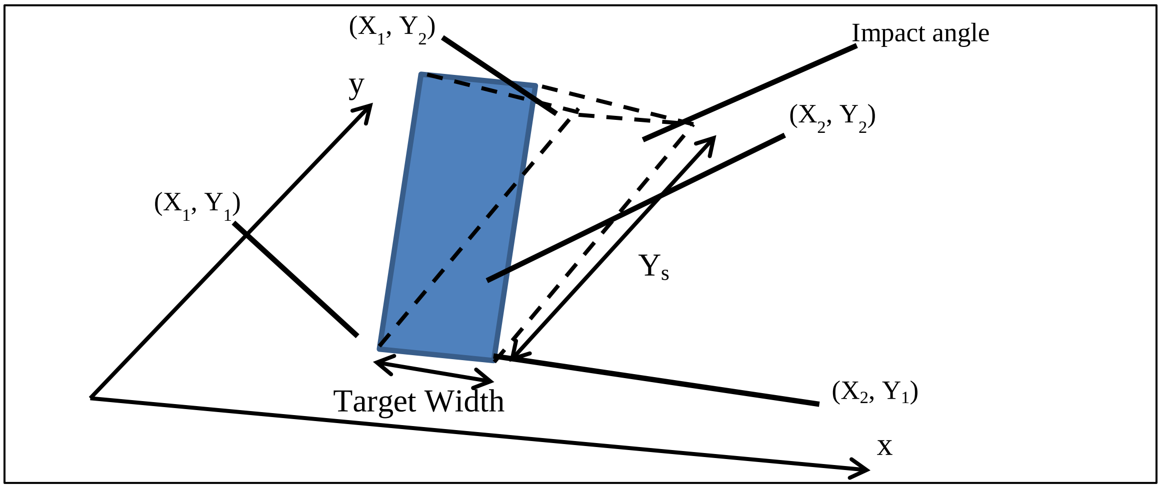


Figure 41. Flow diagram of direct fire weapons module.

$$X_1 = \text{Target Location}_x - \frac{\text{Target Width}}{2} \tag{10.13}$$

$$X_2 = \text{Target Location}_x + \frac{\text{Target Width}}{2} \tag{10.14}$$

$$Y_S = \frac{\text{Height of Target}}{\tan \text{Impact Angle}} \tag{10.15}$$

$$Y_1 = \text{Target Location}_y - \frac{Y_S}{2} \tag{10.16}$$

$$Y_2 = \text{Target Location}_y + \frac{Y_S}{2} \tag{10.17}$$

Effectiveness Calculation for Impact Rounds

With the target area projected on ground, we can then assess whether the aim points have hit any of the targets, where the aim point will lie within the four coordinates of X_1, X_2, Y_1 and Y_2 computed in the previous section. If the aim point has hit the target, a random number between zero and one is generated and compared with an input $P_{Incapacitated}$ to see if the target is incapacitated when it is hit by a bullet. The target is considered to be incapacitated if the generated number is smaller than the provided probability. For example, the target is considered incapacitated if the generated number is 0.7 and $P_{Incapacitated}$ is 0.9. If a target is incapacitated after being hit, the incapacitated due to direct hit counter increases, and the coordinates of the target is changed to 9999 to take it out of the simulation until the next iteration. The effectiveness can be computed as:

$$Incapacitation\ due\ to\ Frag\ Hit\ per\ iteration = \frac{Total\ Indirect\ Incapacitation}{no\ of\ iteration} \tag{10.18}$$

$$PD\ due\ to\ direct\ hit = \frac{Incapacitation\ due\ to\ Frag\ Hit\ per\ iteration}{no\ of\ targets} \tag{10.19}$$

Effectiveness Calculation for Fragmentation Rounds

If the round used is a fragmentation round, a new probability of incapacitation is computed based on the lethal area of the round and the distance of the round from the target, if the round did not achieve any direct hit. This new probability is then compared to a random number to determine if the target is incapacitated by fragmentations. If the generated number is lower than the computed probability, an incapacitation due to

indirect hit is registered and the target is taken out of the simulation by changing its coordinates to 9999.

$Distance, r =$

$$\sqrt{(Aim\ Point_y - \ Target\ Location_y)^2 + (Aim\ Point_y - \ Target\ Location_y)^2} \tag{10.20}$$

$$P_{CAS} = P_{I/H}\exp(\frac{-\pi r^2}{AL}) \tag{10.21}$$

The effectiveness due to fragmentation rounds can then be computed as:

$$Average\ Incapacitation\ due\ to\ Fragmentation\ per\ iteration = \frac{Total\ Direct\ Hit}{no\ of\ iteration} \tag{10.22}$$

$$PD\ due\ to\ fragmentation = \frac{Average\ Incapacitation\ due\ frag\ per\ iteration}{no\ of\ targets} \tag{10.23}$$

C. DIRECT FIRE MODULE IN MATLAB

The direct fire module is modeled as a function using the equations explained in the previous section. The input commands and the associated outputs for the module are depicted in Table 27.

Input	`Direct_Fire_FBAR_Model(Direct_Fire_Data)`
Output	`Direct_Fire_output = [Average_P_D, Proportion_Direct_Hit, Average_P_D_F,Proportion_Indirect_Hit,Total_Average_Inc,Proportion_Inc];`

Table 27. Input and output commands for direct fire module.

Note that the input Direct_Fire_Data is an array. The details of the inputs that form this array can be found in the Appendices.

Note also the following:

1) The program assumes that all the targets are located at a considerable distance where the aim points are uniformly distributed instead of being aimed directly at the targets.

2) The targets are assumed to have uniform height and width.

3) The inputs for the standard deviation in MPI and PE errors are in mils. Hence, there is a need to convert these deviations to feet before they can be used in the Monte Carlo simulation.

4) The vitality of the targets to withstand hit is assumed to be constant.

D. RESULTS OF DIRECT FIRE MODULE

There is no existing equivalent program to verify the results of the module. Hence, a baseline case study was developed to analyze the results when some of the parameters are changed to see if they are reasonable, as shown in Table 28. The results are tabulated in Table 29.

Parameters	Value	Parameters	Value
# of Targets	8	# of Aim Points	8
Target Area Length (ft)	80	Target Area Width (ft)	50
Range to Target (ft)	50	Round Impact Angle of Fall	20
Precision Error (Deflection) (mils)	5	Precision Error (Range) (mils)	5
MPI Error (Deflection) (mils)	5	MPI Error (Range) (mils)	5
TOF deviation (ft)	0	Prob of Incapacitation if Hit	0.85
Lethal Area (ft^2)	20	Minimum Separation Distance (ft)	5
Target Width (ft)	1	Target Height (ft)	6
Round Remaining Velocity (ft/s)	100	Number of Iterations	10000

Table 28. Baseline case study values.

Case Study	Changes in Parameters	PD
1 (Baseline)	No Changes	0.052
2	Lethal Area Increase to $40ft^2$	0.075
3	Lethal Area Reduces to $10ft^2$	0.039
4	MPI and Precision Error Reduces to 1 mils	0.051
5	MPI and Precision Error Increases to 20 mils	0.052
6	Prob of Incapacitation if Hit reduces to 0.75	0.046
7	Prob of Incapacitation if Hit increasess to 0.95	0.059
8	No of iterations increases to 100000	0.053
9	Kill Mode changes from Fragmentation to Impact	0.027
10	# of Aim Points increases to 16	0.092
11	# of Targets increases to 16	0.051
12	Target Area Length reduces to 40, Target Area Width reduces to 25	0.171
13	Target Area Length increases to 160, Target Area Width increases to 100	0.013
14	Range to target increases to 100ft	0.052
15	Range to target reduces to 20ft	0.052

Table 29. Results of direct fire weapons module.

The results seem reasonable based on the following observations:

1) When the lethal area is increased in Case Study 2 and reduced in Case Study 3, the PD increases and decreases, respectively.

2) Changes to MPI and PE errors in Case Study 4 and Case Study 5 do not have a significant effect on the computed PD. This is likely due to a combination of factors such as the range to target, the number of targets and the number of aim points.

3) Changes to Range to Target in Case Study 14 and Case Study 15 has no effect, which was expected considering that changes to MPI and PE errors had negligible effects on the PD.

4) When the probability of incapacitation is reduced to 0.75 in Case Study 6 and increased to 0.95 in Case Study 7, the PD decreases and increases, respectively.

5) When the number of iterations is increased to 100,000 from 10,000 in Case Study 8, the computed PD remains relatively constant increasing from 0.052 to 0.053. Hence, the number of iterations (Qty: 10,000) used to test the results should be representative of the behavior of the Monte Carlo program.

6) When kill mode of projectile is changed to impact from fragmentation, the PD reduces significantly.

7) When the aim point increases to 16 in Case Study 10, the PD increases.

8) Increasing the number of targets to 16 in Case Study 11 has negligible effects. This can be explained by two phenomena expected by the increase. First, the PD should theoretically drop as there are more targets to aim at while the number of aim points remains the same. Conversely, the PD was expected to increase as there are now more targets in the area, thus making it easier for aim point to actually hit a target. Hence, these two contradictory factors caused a negligible change in PD when the number of targets is changed.

9) When the target area is reduced in Case Study 12 and increased in Case Study 13, the PD increases and reduces, respectively.

XI. CONCLUSION AND RECOMMENDATIONS

In conclusion, the weapons system effectiveness program was successfully developed. It provides an integrated program which allows the damage computation of both AS and SS weapons, while incorporating a high-fidelity trajectory module to compute the impact condition of the weapons, thus increasing the fidelity of the program. This research serves as a useful tool for students learning the computation methodology of weapons system effectiveness, as well as for use by countries with no clearance to use an existing unclassified weapons effectiveness program. The program addresses most of the conventional AS and SS weapons.

The results from the test have been favorable, supporting the accuracy of the results from the program. The majority of the test cases were compared with existing Excel programs validated with established programs such as the JWM, with the exception of the two Monte Carlo simulations for SS weapons. Nevertheless, test cases have been run to ensure that the values computed by these two Monte Carlo modules are reasonable.

There can be improvements made to the program subsequently by increasing the scope of the program. Additional modules can be developed to compute the weapon effectiveness against specific target types such as bridges, underground bunkers and tunnels, some of which require the usage of advanced penetrator weapons. Another area of interest is in collateral damage estimation, which has generated substantial human rights concern and can affect mission planning on the type of weapons that can be deployed during Operations Other than War (OOTW).

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. USER INPUTS IN EXCEL GUI

A. INPUTS FOR TRAJECTORY MODULE

S/No	Field	Remarks
1	System Type	Choose “Air-to-Surface”
Air-to-Surface Systems		
2	Program Type	Choose “Weapon Trajectory”
3	Warhead Size	Choose warhead size from dropdown list. Do not choose cluster bombs and rockets.
Aircraft Release Data		
4	Aircraft Speed	Enter aircraft speed.
5	Release Altitude	Enter altitude that weapon is released from.
6	Dive Angle	Enter dive angle of the aircraft.
7	Ejection Velocity	Enter the ejection velocity that weapon is released from aircraft.

B. INPUT FOR WEAPONS ACCURACY MODULE

S/No	Field	Remarks
1	System Type	Choose “Air-to-Surface”
Air-to-Surface Systems		
2	Program Type	Choose “Weapon Accuracy”
3	Warhead Size	Choose warhead size from dropdown list. Do not choose cluster bombs and rockets.
4	Aircraft Type	Choose aircraft type which affects aircraft sensor parameters.
Aircraft Release Data		
5	Aircraft Speed	Enter aircraft speed.
6	Release Altitude	Enter altitude that weapon is released from.
7	Dive Angle	Enter dive angle of the aircraft.

S/No	Field	Remarks
8	Ejection Velocity	Enter the ejection velocity that weapon is released from aircraft.
Weapon Accuracy Data Requirement		
9	Bomb Mode	Choose “CCRP,” “CCIP” or “BOC.”
10	Range to Target	To enter range to target value if CCRP is chosen in Bomb Mode.
11	Targeting Pod	To choose Targeting Pod if CCRP is chosen in Bomb Mode.
12	Target Position Error	To choose “Default,” “GPS Coordinate” or “Self-Input” if BOC is chosen in Bomb Mode.
13	Xt	If “Self-Input is chosen in Target Position Error, to enter error in x-axis, y-axis and z-axis.
14	Yt	
15	Zt	
Weapons Parameters		
16	Ballistic Dispersion	To enter the ballistic dispersion of the weapon in mils.

C. INPUT FOR SINGLE WEAPON UNITARY TARGET MODULE

S/No	Field	Remarks
1	System Type	Choose “Air-to-Surface”
Air-to-Surface Systems		
2	Program Type	Choose “Target Damage Computation for Bombs”
3	Weapon Type	Choose “Unguided,” “Guided(LGB)” or “Guided(GPS)”
4	Warhead Size	Choose warhead size from dropdown list. Do not choose cluster bombs and rockets.
5	# of Weapons	Choose “Single.”
6	Target Type	Choose “Single.”
7	Aircraft Type	Choose aircraft type which affects aircraft sensor parameters.
Aircraft Release Data		
8	Aircraft Speed	Enter aircraft speed.

S/No	Field	Remarks
9	Release Altitude	Enter altitude that weapon is released from.
10	Dive Angle	Enter dive angle of the aircraft.
11	Ejection Velocity	Enter the ejection velocity that weapon is released from aircraft.
Additional Target Damage Computation Data Requirement		
12	Kill Mode	Choose “Blast” or “Fragmentation.”
13	MAE	Enter MAE value.
14	CEP/REP/DEP Input	Choose “Manual (REP/DEP),” “Manual (CEP)” or “Auto” if “Unguided” is selected in Weapon Type.
15	CEP Input	Choose “Manual” or “Default” if “Guided(LGB)”or “Guided(GPS)” is selected in Weapon Type.
16	CEP	To Enter CEP value of Manual Input of CEP is chosen in either CEP/REP/DEP Input or CEP Input.
17	REP	To Enter REP and DEP value of Manual Input of REP/DEP is chosen in either CEP/REP/DEP Input.
18	DEP	
19	Impact Angle Input	Choose “Initial Harp Angle,” “Manual Input” or “Trajectory Program” if “Guided (LGB)” is selected in Weapon Type.
20	Impact Angle	Enter Impact Angle if “Manual Input” is selected in Impact Angle Input.
21	Miss/Prob Hit Input	Choose “Manual” or “Default” if “Guided(LGB)”or “Guided(GPS)” is selected in Weapon Type.
22	Prob Near Miss	Enter probability value (Between 0 and 1) if “Manual” is chosen in Miss/Prob Hit Input.
23	Prob Hit	
Weapon Accuracy Data Requirement		
24	Bomb Mode	Choose “CCRP,” “CCIP” or “BOC” if “Unguided” is selected in Weapon Type.
25	Range to Target	To enter range to target value if CCRP is chosen in Bomb Mode.
26	Targeting Pod	To choose Targeting Pod if CCRP is chosen in Bomb Mode.

S/No	Field	Remarks
27	Target Position Error	To choose “Default,” “GPS Coordinate” or “Self-Input” if BOC is chosen in Bomb Mode, or “Guided(GPS)” is selected in Weapon Type and “Default” is selected in CEP input.
28	Xt	If “Self-Input is chosen in Target Position Error, to enter error in x-axis, y-axis and z-axis.
29	Yt	
30	Zt	
Weapons Parameters		
31	Reliability	To enter reliability of the weapon
32	Ballistic Dispersion	To enter the ballistic dispersion of the weapon in mils.

D. INPUT FOR SINGLE WEAPON AREA TARGET MODULE

S/No	Field	Remarks
1	System Type	Choose “Air-to-Surface”
Air-to-Surface Systems		
2	Program Type	Choose “Target Damage Computation for Bombs”
3	Weapon Type	Choose “Unguided,” “Guided(LGB)” or “Guided(GPS)”
4	Warhead Size	Choose warhead size from dropdown list. Do not choose cluster bombs and rockets.
5	# of Weapons	Choose “Single.”
6	Target Type	Choose “Multiple.”
7	Target Length	Enter Target Length/Width if “Multiple” is chosen in “Target Type.”
8	Target Width	
9	Aircraft Type	Choose aircraft type which affects aircraft sensor parameters.
Aircraft Release Data		
10	Aircraft Speed	Enter aircraft speed.
11	Release Altitude	Enter altitude that weapon is released from.

S/No	Field	Remarks
12	Dive Angle	Enter dive angle of the aircraft.
13	Ejection Velocity	Enter the ejection velocity that weapon is released from aircraft.
Additional Target Damage Computation Data Requirement		
14	Kill Mode	Choose “Blast” or “Fragmentation.”
15	MAE	Enter MAE value.
16	CEP/REP/DEP Input	Choose “Manual (REP/DEP),” “Manual (CEP)” or “Auto” if “Unguided” is selected in Weapon Type.
17	CEP Input	Choose “Manual” or “Default” if “Guided(LGB)”or “Guided(GPS)” is selected in Weapon Type.
18	CEP	To Enter CEP value of Manual Input of CEP is chosen in either CEP/REP/DEP Input or CEP Input.
19	REP	To Enter REP and DEP value of Manual Input of REP/DEP is chosen in either CEP/REP/DEP Input.
20	DEP	
21	Impact Angle Input	Choose “Initial Harp Angle,” “Manual Input” or “Trajectory Program” if “Guided (LGB)” is selected in Weapon Type.
22	Impact Angle	Enter Impact Angle if “Manual Input” is selected in Impact Angle Input.
23	Miss/Prob Hit Input	Choose “Manual” or “Default” if “Guided(LGB)”or “Guided(GPS)” is selected in Weapon Type.
24	Prob Near Miss	Enter probability value (Between 0 and 1) if “Manual” is chosen in Miss/Prob Hit Input.
25	Prob Hit	
Weapon Accuracy Data Requirement		
26	Bomb Mode	Choose “CCRP,” “CCIP” or “BOC” if “Unguided” is selected in Weapon Type.
27	Range to Target	To enter range to target value if CCRP is chosen in Bomb Mode.
28	Targeting Pod	To choose Targeting Pod if CCRP is chosen in Bomb Mode.
29	Target Position Error	To choose “Default,” “GPS Coordinate” or “Self-Input” if BOC is

S/No	Field	Remarks
		chosen in Bomb Mode, or “Guided(GPS)” is selected in Weapon Type and “Default” is selected in CEP input.
30	Xt	If “Self-Input is chosen in Target Position Error, to enter error in x-axis, y-axis and z-axis.
31	Yt	
32	Zt	
Weapons Parameters		
33	Reliability	To enter reliability of the weapon
34	Ballistic Dispersion	To enter the ballistic dispersion of the weapon in mils.

E. INPUT FOR STICK DELIVERY MODULE

S/No	Field	Remarks
1	System Type	Choose “Air-to-Surface”
Air-to-Surface Systems		
2	Program Type	Choose “Target Damage Computation for Bombs”
3	Weapon Type	Choose “Unguided”
4	Warhead Size	Choose warhead size from dropdown list. Do not choose cluster bombs and rockets.
5	# of Weapons	Choose “Multiple.”
6	Qty	Enter qty of weapons dropped when “Multiple” is chosen for # of Weapons.
7	Aircraft Type	Choose aircraft type which affects aircraft sensor parameters.
8	Target Length	Enter Target Length/Width.
9	Target Width	
Aircraft Release Data		
10	Aircraft Speed	Enter aircraft speed.
11	Release Altitude	Enter altitude that weapon is released from.
12	Dive Angle	Enter dive angle of the aircraft.

S/No	Field	Remarks
13	Ejection Velocity	Enter the ejection velocity that weapon is released from aircraft.
Additional Target Damage Computation Data Requirement		
14	Kill Mode	Choose “Blast” or “Fragmentation.”
15	Damage Function Type	
16	MAE	Enter MAE value.
17	CEP/REP/DEP Input	Choose “Manual (REP/DEP),” “Manual (CEP)” or “Auto” if “Unguided” is selected in Weapon Type.
18	CEP Input	Choose “Manual” or “Default” if “Guided(LGB)”or “Guided(GPS)” is selected in Weapon Type.
19	CEP	To Enter CEP value of Manual Input of CEP is chosen in either CEP/REP/DEP Input or CEP Input.
20	REP	To Enter REP and DEP value of Manual Input of REP/DEP is chosen in either CEP/REP/DEP Input.
21	DEP	
22	Impact Angle Input	Choose “Initial Harp Angle,” “Manual Input” or “Trajectory Program” if “Guided (LGB)” is selected in Weapon Type.
23	Impact Angle	Enter Impact Angle if “Manual Input” is selected in Impact Angle Input.
24	Miss/Prob Hit Input	Choose “Manual” or “Default” if “Guided(LGB)”or “Guided(GPS)” is selected in Weapon Type.
25	Prob Near Miss	Enter probability value (Between 0 and 1) if “Manual” is chosen in Miss/Prob Hit Input.
26	Prob Hit	
Weapon Accuracy Data Requirement		
27	Bomb Mode	Choose “CCRP,” “CCIP” or “BOC” if “Unguided” is selected in Weapon Type.
28	Range to Target	To enter range to target value if CCRP is chosen in Bomb Mode.
29	Targeting Pod	To choose Targeting Pod if CCRP is chosen in Bomb Mode.
30	Target Position Error	To choose “Default,” “GPS Coordinate” or “Self-Input” if BOC is

S/No	Field	Remarks
		chosen in Bomb Mode, or "Guided(GPS)" is selected in Weapon Type and "Default" is selected in CEP input.
31	Xt	If "Self-Input is chosen in Target Position Error, to enter error in x-axis, y-axis and z-axis.
32	Yt	
33	Zt	
Additional Stick Delivery Requirement		
34	# of Release Pulses	Enter number of release pulses
35	# Weapons/Pulses	Choose "1" or "2" to select the number of weapons that will be released with each release pulse.
36	Intervalometer Settings	Choose "Time" or "Distance" to specify how the intervalometer is set. This option is not available if dive angle is not equal to zero.
37	Time	Enter the time interval between each pulse if "Time" is selected for intervalometer setting or dive angle is not equal to zero.
38	Distance	Enter the distance interval between each pulse if "Distance" is selected for intervalometer setting. This option is not available if dive angle is not equal to zero.
39	Stick Width	Enter the stick width.
Weapons Parameters		
40	Reliability	To enter reliability of the weapon
41	Ballistic Dispersion	To enter the ballistic dispersion of the weapon in mils.

F. INPUT FOR CLUSTER WEAPONS MODULE

S/No	Field	Remarks
1	System Type	Choose "Air-to-Surface"
Air-to-Surface Systems		
2	Program Type	Choose "Target Damage Computation for Bombs"
3	Weapon Type	Choose "Cluster"

S/No	Field	Remarks
4	Warhead Size	Choose “Cluster Munitions”
5	# of Weapons	Choose “Single” or “Multiple”
6	Qty	Enter qty of weapons dropped when “Multiple” is chosen for # of Weapons.
7	Aircraft Type	Choose aircraft type which affects aircraft sensor parameters.
8	Target Length	Enter Target Length/Width.
9	Target Width	
Aircraft Release Data		
10	Aircraft Speed	Enter aircraft speed.
11	Release Altitude	Enter altitude that weapon is released from.
12	Dive Angle	Enter dive angle of the aircraft.
13	Ejection Velocity	Enter the ejection velocity that weapon is released from aircraft.
Additional Target Damage Computation Data Requirement		
14	Kill Mode	Choose “Blast” or “Fragmentation.”
15	MAE	Enter MAE value.
16	CEP/REP/DEP Input	Choose “Manual (REP/DEP),” “Manual (CEP)” or “Auto” if “Unguided” is selected in Weapon Type.
17	CEP	To Enter CEP value if Manual Input of CEP is chosen in CEP/REP/DEP.
18	REP	To Enter REP and DEP value of Manual Input of REP/DEP is chosen in either CEP/REP/DEP Input.
19	DEP	
Weapon Accuracy Data Requirement		
20	Bomb Mode	Choose “CCRP,” “CCIP” or “BOC” if “Unguided” is selected in Weapon Type.
21	Range to Target	To enter range to target value if CCRP is chosen in Bomb Mode.
22	Targeting Pod	To choose Targeting Pod if CCRP is chosen in Bomb Mode.

S/No	Field	Remarks
23	Target Position Error	To choose “Default,” “GPS Coordinate” or “Self-Input” if BOC is chosen in Bomb Mode.
24	Xt	If “Self-Input is chosen in Target Position Error, to enter error in x-axis, y-axis and z-axis.
25	Yt	
26	Zt	
Additional Stick Delivery Requirement (Option will appear only if # of Weapons selected is “Multiple”)		
27	# of Release Pulses	Enter number of release pulses
28	# Weapons/Pulses	Choose “1” or “2” to select the number of weapons that will be released with each release pulse.
29	Intervalometer Settings	Choose “Time” or “Distance” to specify how the intervalometer is set. This option is not available if dive angle is not equal to zero.
30	Time	Enter the time interval between each pulse if “Time” is selected for intervalometer setting or dive angle is not equal to zero.
31	Distance	Enter the distance interval between each pulse if “Distance” is selected for intervalometer setting. This option is not available if dive angle is not equal to zero.
32	Stick Width	Enter the stick width.
Additional Cluster Munitions Data Requirement		
33	# of submunitions	Enter number of submunitions in cluster weapon.
34	Submunitions Reliability	Enter reliability of a submunition.
35	Pattern Type	Select “Length/Width” or “Radius.”
36	Single Dispenser Pattern Radius	To enter pattern radius if “Radius” is selected in Pattern Type.
37	Single Dispenser Pattern Length	To enter pattern length and width if “Length/Width” is selected in Pattern Type.
38	Single Dispenser Pattern Width	

S/No	Field	Remarks
39	Release Type	Select “Functioning Time” or “Functioning Altitude.”
40	Functioning Time	Enter the time delay before submunitions are released if “Functioning Time” is selected in release type.
41	Functioning Altitude	Enter the altitude in which the submunitions are released if “Functioning Altitude” is selected in release type.
Weapons Parameters		
42	Reliability	To enter reliability of the cluster weapon dispenser.
43	Ballistic Dispersion	To enter the ballistic dispersion of the weapon in mils.

G. INPUT FOR PROJECTILE MODULE

S/No	Field	Remarks
1	System Type	Choose “Air-to-Surface”
Air-to-Surface Systems		
2	Program Type	Choose “Target Damage Computation for Rockets/Projectiles”
3	Rounds Fired	Enter number of rounds fired.
4	Hits to Kill Target	Enter number of hits required to kill the target
Aircraft Release Data		
5	Aircraft Speed	Enter aircraft speed.
6	Release Altitude	Enter altitude that weapon is released from.
7	Dive Angle	Enter dive angle of the aircraft.
Additional Target Damage Computation Data Requirement		
8	Kill Mode	Choose “Blast” or “Fragmentation.”
9	Damage Function Type	Choose “MAE(Blast),” “MAE(Fragmentation)” or “Av.”
10	MAE	Enter MAE value.
11	CEP/REP/DEP Input	Choose “REP/DEP” or “CEP.”
12	CEP	To Enter CEP value if “CEP” is chosen in CEP/REP/DEP Input.

S/No	Field	Remarks
13	REP	To Enter REP and DEP if "REP/DEP" is chosen in CEP/REP/DEP Input.
14	DEP	
Weapons Parameters		
15	Reliability	To enter reliability of the weapon
16	Ballistic Dispersion	To enter the ballistic dispersion of the weapon in mils.

H. INPUT FOR INDIRECT FIRE WEAPON AGAINST AREA TARGET MODULE

S/No	Field	Remarks
1	System Type	Choose "Surface-to-Surface."
Surface-to-Surface Systems		
2	Program Type	Choose "Indirect Fire."
3	Target Type	Choose "Area."
4	Weapon Type	Choose "ICM" or "Unitary" when "Area" is selected in Target Type.
5	Damage Function Type	Choose "MAE (Blast)" or "MAE (Fragmentation)" when "Unitary" is selected in Weapon Type.
6	# Rounds / Volley	Enter the number of rounds that is fired in each volley.
7	# Volley Fired	Enter the numbers of volley fired.
8	# aim points in range direction	Enter the number of aim points in the range direction.
9	# aim points in deflection direction	Enter the number of aim points in the deflection direction.
10	Target Dimension Type	Choose "Radius" or "Length/Width."
11	Target Radius	Enter radius of target dimension when "Radius" is chosen in Target Dimension Type.
12	Target Length	Enter length and width of target dimension when "Length" and "Width" is chosen in Target Dimension Type.
13	Target Width	

S/No	Field	Remarks
Accuracy Computation		
14	Precision Error Range	Enter the precision error in range.
15	Precision Error Deflection	Enter the precision error in deflection.
16	MPI Error Range	Enter the MPI error in range.
17	MPI Error Deflection	Enter the MPI error in deflection.
18	TLE	Enter the Target Location Error
Round Parameters		
19	Round Impact Angle of Fall	Enter the round impact angle when “Unitary” is selected for Weapon Type.
20	Round Reliability	Enter reliability of round between 0 and 1.
21	Single Round Lethal Area	Enter Lethal area for round for unitary warhead and submunition for ICM warhead.
22	Pattern Adjustment Factor	Enter Pattern Adjustment Factor
23	Single Round Pattern Dimension	Select “Length/Width” or “Radius” when “ICM” is selected for Weapon Type.
24	Single Round Pattern Length	Enter the pattern length and width of the submunitions when “Length/Width” is selected.
25	Single Round Pattern Width	
26	Single Round Pattern Radius	Enter the pattern radius of the submunitions when “radius” is selected.
27	Submunitions Reliability	Enter reliability of the submunitions when “ICM” is entered in Weapon Type.
28	# of Submunitions per Round	Enter the number of submunitions in each round when “ICM” is entered in Weapon Type.

I. INPUT FOR INDIRECT FIRE WEAPON AGAINST POINT TARGET MODULE

S/No	Field	Remarks
1	System Type	Choose "Surface-to-Surface."
Surface-to-Surface Systems		
2	Program Type	Choose "Indirect Fire."
3	Target Type	Choose "Single."
4	Damage Function Type	Choose "MAE (Blast)" or "MAE (Fragmentation)."
5	# Rounds / Volley	Enter the number of rounds that is fired in each volley.
6	# Volley Fired	Enter the numbers of volley fired.
Accuracy Computation		
7	Precision Error Range	Enter the precision error in range.
8	Precision Error Deflection	Enter the precision error in deflection.
9	MPI Error Range	Enter the MPI error in range.
10	MPI Error Deflection	Enter the MPI error in deflection.
11	TLE	Enter the Target Location Error
Round Parameters		
12	Single Round Lethal Area	Enter Lethal area for round for unitary warhead and submunition for ICM warhead.
Monte Carlo Simulation Data		
13	Number of Iterations	Enter the number of iterations required for the Monte Carlo Simulation.

J. INPUT FOR DIRECT FIRE WEAPON MODULE

S/No	Field	Remarks
1	System Type	Choose "Surface-to-Surface."
Surface-to-Surface Systems		

S/No	Field	Remarks
2	Program Type	Choose “Direct Fire.”
3	# of Targets	Enter number of targets.
4	#of Aim Points	Enter number of aim points
5	Target Length	Enter length and width of the area where target is residing.
6	Target Width	
7	Range to Target	Enter the range from firing entity to target.
Accuracy Computation		
8	Precision Error Range	Enter the precision error in range.
9	Precision Error Deflection	Enter the precision error in deflection.
10	MPI Error Range	Enter the MPI error in range.
11	MPI Error Deflection	Enter the MPI error in deflection.
12	TOF Deviation	Enter the deviation for TOF.
Round Parameters		
13	Single Round Lethal Area	Enter Lethal area for each round when “Fragment” is chosen for Damage Function.
14	Round Impact Angle of Fall	Enter the round impact angle.
15	Probability of Incapacitation if Hit	Enter the probability of incapacitation if the target is hit.
FBAR Direct Fire Computation		
16	Target Width	Enter width of targets.
17	Target Height	Enter height of targets.
18	Minimum Separation	Enter the minimum separation distance between each target.
19	Round Remaining Velocity	Enter the round remaining velocity to compute variation in muzzle velocity.
20	Damage Function	Choose “Fragment” or “Impact.”

S/No	Field	Remarks
Monte Carlo Simulation Data		
21	Number of Iterations	Enter the number of iterations required for the Monte Carlo Simulation.

APPENDIX B. ARRAY INPUTS TO MATLAB

A. ARRAYS FOR AIR-TO-SURFACE SYSTEMS

1. Initial Data

Initial Data		
Parameter	**Data Representation**	**MATLAB S/No**
Program Type	N.A.	1
Warhead Size	See 5 Weapon Data Sheet	2
Weapon Type	1: Guided(LGB) 2: Guided(GPS) 3: Unguided 4: Cluster	3
# of Weapons	1: Single 2: Multiple	4
Target Type	1: Single 2: Multiple	5
Airspeed	N.A.	6
Release Altitude	N.A.	7
Dive Angle	N.A.	8
Ejection Velocity	N.A.	9

2. Weapon Accuracy Data Requirement

Weapon Accuracy Data Requirement		
Parameter	**Data Representation**	**MATLAB S/No**
Weapon Accuracy Bomb Mode	1: CCRP 2: CCIP 3: BOC	1
Range to Target	N.A.	2
Targetting Pod	1: 2nd Gen Pod/Default 2: 3rd Gen Pod	3
Target Position Error	N.A.	4
Xt	N.A.	5
Yt	N.A.	6
Zt	N.A.	7
Ballistic Error	N.A.	8

Weapon Accuracy Data Requirement		
Parameter	**Data Representation**	**MATLAB S/No**
Aircraft Type	1: Default (Fighter) 2: Fighter 3: Bomber 4: Rotary Wings	9

3. Single Weapon Unitary Target

Single Weapon Unitary Target		
Parameter	**Data Representation**	**MATLAB S/No**
MAE Input	1: Self-Input 2: Auto-Input	1
MAE	N.A.	2
Kill Mode	1: Blast 2: Fragmentation	3
Pk Hit/Miss Input (Guided)	1: Manual 2: Default(Based on Guidance Type)	4
CEP/REP/DEP Input	1: Manual (CEP) 2: Manual (REP/DEP) 3: Weapon Accuracy Program (Auto)	5
CEP	N.A.	6
REP	N.A.	7
DEP	N.A.	8
Weapon Reliability	N.A.	9
Impact Angle Type	1: Manual Input 2: Trajectory Program (LGB&Unguided) 3: Dive Angle (LGB only)	10
Impact Angle	N.A.	11
Ballistic Error	N.A.	12
Prob Near Miss	N.A.	13
Prob Hit	N.A.	14
CEP Input (Guided)	1: Manual 2: Default(Based on Guidance	15

Single Weapon Unitary Target		
Parameter	**Data Representation**	**MATLAB S/No**
	Type)	

4. **Single Weapon Area Target**

Single Weapon Area Target		
Parameter	**Data Representation**	**MATLAB S/No**
LA	N.A.	1
WA	N.A.	2

5. **Stick Delivery**

Stick Delivery		
Parameter	**Data Representation**	**MATLAB S/No**
# of Release Pulses		1
# Weapons / Release Pulses		2
Intervalometer Setting	1: Time 2: Distance	3
Time		4
Distance		5
Stick Width		6

6. **Cluster Munitions**

Cluster Munitions		
Parameter	**Data Representation**	**MATLAB S/No**
LA	N.A.	1
WA	N.A.	2
Number of Submunitions	N.A.	3
Release Type	1: Time 2: Distance	4

Cluster Munitions		
Parameter	**Data Representation**	**MATLAB S/No**
Functioning Time	N.A.	5
Functioning Altitude	N.A.	6
Pattern Type	1: Lenth/Width 2: Radius	7
Single Dispenser Pattern Length	N.A.	8
Single Dispenser Pattern Width	N.A.	9
Single Dispenser Pattern Radius	N.A.	10
Submunition Reliabiity	N.A.	11
Dispenser Reliabiity	N.A.	12
# of Release Pulses	N.A.	13
# Weapons / Release Pulses	N.A.	14
Intervalometer Setting	1: Time 2: Distance	15
Time	N.A.	16
Distance	N.A.	17
Stick Width	N.A.	18
Weapon Accuracy Bomb Mode	1: CCRP 2: CCIP 3: BOC	19
Range to Target	N.A.	20
Targetting Pod	1: 2nd Gen Pod/Default 2: 3rd Gen Pod	21
Target Position Error	N.A.	22
Xt	N.A.	23
Yt	N.A.	24
Zt	N.A.	25
Ballistic Error	N.A.	26
Aircraft Type	1: Default (Fighter) 2: Fighter 3: Bomber 4: Rotary Wings	27

Cluster Munitions		
Parameter	**Data Representation**	**MATLAB S/No**
CEP/REP/DEP Input	1: Manual (CEP) 2: Manual (REP/DEP) 3: Weapon Accuracy Program (Auto)	28
CEP	N.A.	29
REP	N.A.	30
DEP	N.A.	31

7. Rockets/Projectiles

Rockets/Projectiles		
Parameter	**Data Representation**	**MATLAB S/No**
MAE	N.A.	1
Kill Mode	1: Blast 2: Fragmentation	2
Damage Function Type	1: MAE(Blast) 2: MAE(Frag) 3: Av	3
Rounds Fired	38	4
Hits to Kill Target	1	5
Ballistic Error	N.A.	6
Weapon Reliability	N.A.	7
CEP/REP/DEP Input	1: Manual (CEP) 2: Manual (REP/DEP)	8
CEP	N.A.	9
REP	N.A.	10
DEP	N.A.	11

B. ARRAYS FOR SURFACE-TO-SURFACE SYSTEMS

1. Initial Data

Initial Data		
Parameter	**Data Representation**	**MATLAB S/No**
Program Type	1: Indirect Fire 2: Direct Fire	1
Weapon Type	1: Unitary 2: ICM	2
Damage Function Type	1: MAE(Blast) 2: MAE(Frag)	3
# of Rnds/Volley	1: Single 2: Multiple	4
# Volleys Fired	1: Single 2: Multiple	5
# aim points in range direction	N.A.	6
# aim points in deflection direction	N.A.	7
Target Type	1: Single 2: Area	8
Dimension Type	1: Length/Width 2: Radius	9
Target Length	N.A.	10
Target Width	N.A.	11
Target Radius	N.A.	12
Range from Wpn to Target	N.A.	13

2. Accuracy Computation

Accuracy Computation		
Parameter	**Data Representation**	**MATLAB S/No**
Precision Error Range	N.A.	1
Precision Error Deflection	N.A.	2
MPI Error Range	N.A.	3
MPI Error Deflection	N.A.	4

Accuracy Computation		
Parameter	**Data Representation**	**MATLAB S/No**
TLE	N.A.	5
Monte Carlo Iterations	N.A.	6

3. Round Parameters

Round Parameters		
Parameter	**Data Representation**	**MATLAB S/No**
Round Impact Angle of fall	N.A.	1
Round Reliability	N.A.	2
Single Round Lethal Area	N.A.	3
Pattern Adjustment Factor	N.A.	4
Single Round Pattern Length	N.A.	5
Single Round Pattern Width	N.A.	6
Submunition Reliability	N.A.	7
#Submunitions / Round	N.A.	8

4. Round Parameters

Direct Fire Parameters		
Parameter	**Data Representation**	**MATLAB S/No**
Damage Function Type	1: Impact 2: Fragment	1
Target Area Length	N.A.	2
Target Area Width	N.A.	3
# of Targets	N.A.	4
Minimum Distance	N.A.	5
Target Width	N.A.	6
Height	N.A.	7
# of Aim Point	N.A.	8
Round Remaining Velocity	N.A.	9
Impact Angle	N.A.	10
Lethal Area	N.A.	11
Prob of incapacitation	N.A.	12
Range to target	N.A.	13
Sigma MPI-x	N.A.	14
Sigma MPI-z	N.A.	15
Sigma Precision -x	N.A.	16
Sigma Precision -z	N.A.	17
Sigma TOF	N.A.	18
Iterations	N.A.	19

APPENDIX C. USING THE WEAPONS EFFECTIVENESS PROGRAM

A. INITIAL SETUP

1. Step 1: Copy and paste all related files in a single folder

The first step is to ensure all the relevant files are included and copy and paste them in a single folder. Any file missing or not in the folder will result in an error as the program will not be able to pull the functions it needs.

The files that should be present are as follows:

1) BOC_Model.m

2) CCIP_Model.m

3) CCRP_Model.m

4) clear_excel_file.m

5) Cluster_Munitions_Model.m

6) constant_table.m

7) Direct_Fire_FBAR_Model.m

8) High_Fidelity_Trajectory_Cluster_Munitions_Model.m

9) High_Fidelity_Trajectory_Model.m

10) High_Fidelity_Trajectory_stick_delivery_Model.m

11) Indirect_Monte_Carlo_Model.m

12) Indirect_Weapons_Model.m

13) Main_Excel_Interface.m

14) Rocket_Projectile_Model.m

15) Sensor_Variable_Model.m

16) Single_Wpn_Area_Target_Model.m

17) Single_Wpn_Unitary_Target_Model.m

18) Stick_Delivery_Model.m

19) Weapon_Parameter_Model.m

20) Weaponeering_Program_Main_User_Interface.xlsm

21) Weaponeering_Program_Results.xlsx

22) Write_to_excel_Cluster_Results.m

23) Write_to_excel_Rockets_Projectiles.m

24) Write_to_excel_Single_Target_Unitary_Weapon_Results.m

25) Write_to_excel_Single_Weapon_Area_Target_Results.m

26) Write_to_excel_Stick_Delivery_Results.m

27) Write_to_excel_Superquickie_Results.m

28) Write_to_excel_Trajectory_Results.m

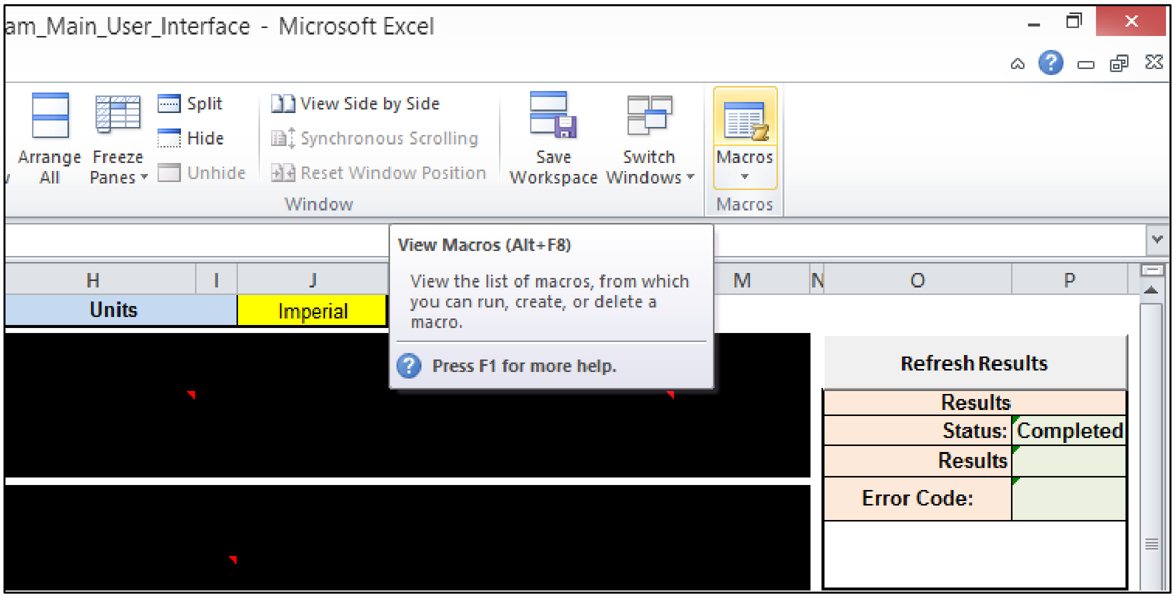
29) Write_to_excel_Weapon_Accuracy_Results.m

30) Zero_Drag_Trajectory_Model.m

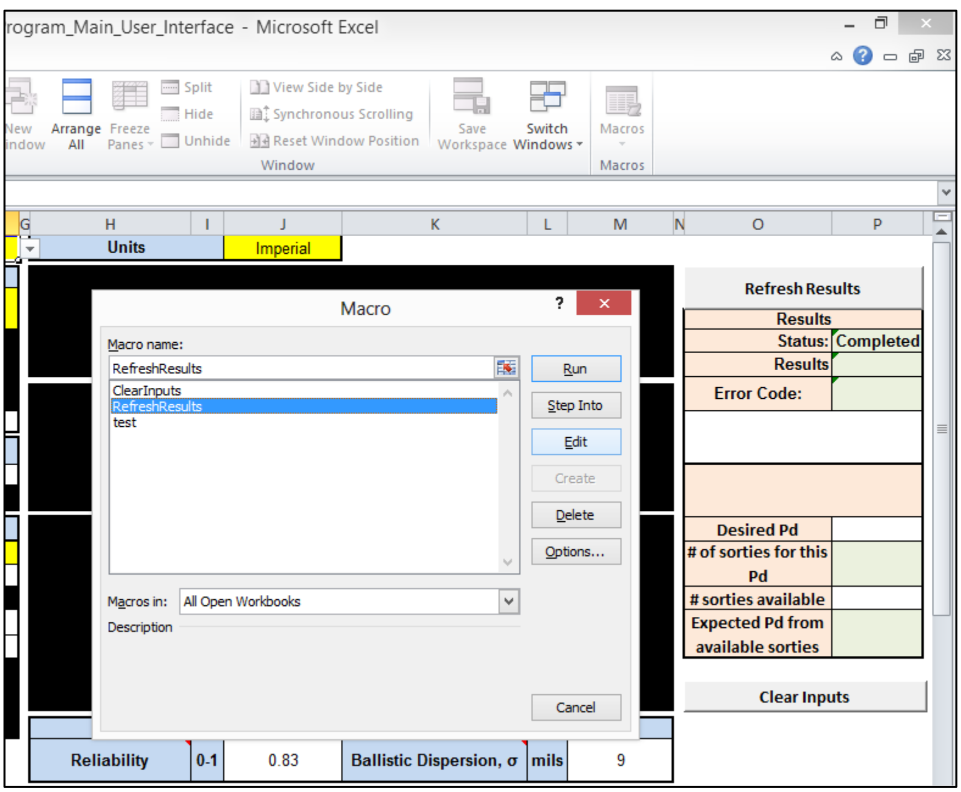
2. Step 2: Modify the Path Directory in Excel MACROS

The path specified in the macros has to be modified and point to the folder that the weapon effectiveness program is being located, in order for the refresh button to work. A step-by-step illustration is provided below.

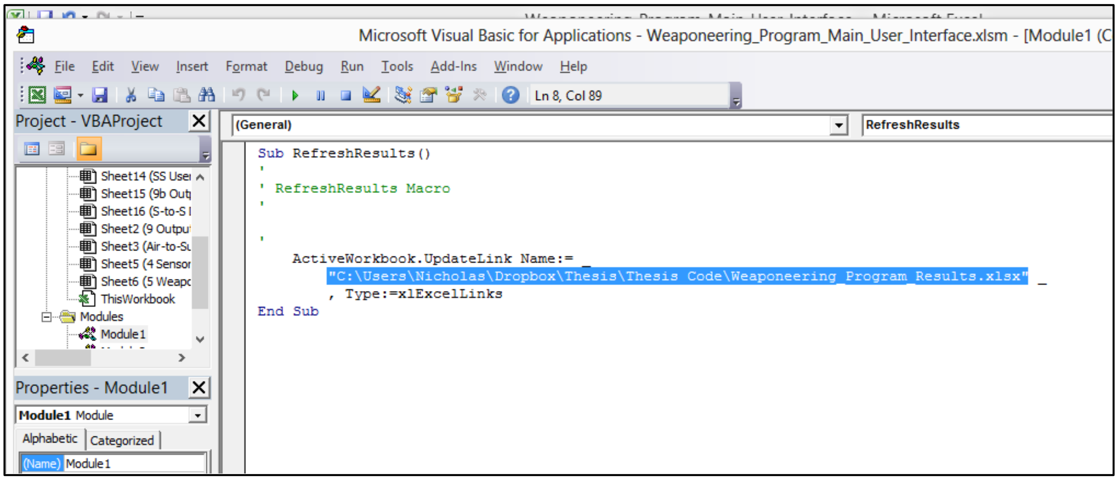
1. Click on the Macros Option.



2. Select RefreshResults and click on Edit.



3. Amend the directory where the weapons effectiveness program is located in the highlighted command. Save after the path is modified.



B. RUNNING THE WEAPON EFFECTIVENESS PROGRAM

The two essential files that have to be opened are as follows:

1) Weaponeering_Program_Main_User_Interface.xlsm Excel file where the user can input the initial release/firing condition.

2) Main_Excel_Interface.m file in Matlab, in which the user must select the run button to run the program.

The results can be viewed in the first file without requiring the user to close the file. A refresh button is provided to update the computed results after the simulation is completed. The refresh button must be pressed manually.

Note: Continuously pressing the refresh button might cause an error when the Excel file attempts to update the results for the Excel results file simultaneously when MATLAB is writing the results into the Excel results file. Under such a circumstance, the program has to be run again.

LIST OF REFERENCES

[1] M. R. Driels, *Weaponeering: Conventional Weapon System Effectiveness* (2nd ed.). Reston, VA: AIAA, 2013.

[2] M. R. Driels, “Zero-Drag Trajectory Impact Condition Computation Datasheet,” Department of Mechanical Engineering, Naval Postgraduate School, unpublished Excel spreadsheet.

[3] M. R. Driels, “Single Weapon Unitary Target Weapon Effectiveness Computation Datasheet,” Department of Mechanical Engineering, Naval Postgraduate School, unpublished Excel spreadsheet.

[4] M. R. Driels, “Single Weapon Area Target Weapon Effectiveness Computation Datasheet,” Department of Mechanical Engineering, Naval Postgraduate School, unpublished Excel spreadsheet.

[5] M. R. Driels, “Stick Delivery Weapon Effectiveness Computation Datasheet,” Department of Mechanical Engineering, Naval Postgraduate School, unpublished Excel spreadsheet.

[6] M. R. Driels, “Cluster Munitions Weapon Effectiveness Computation Datasheet,” Department of Mechanical Engineering, Naval Postgraduate School, unpublished Excel spreadsheet.

[7] M. R. Driels, “Projectiles Weapon Effectiveness Computation Datasheet,” Department of Mechanical Engineering, Naval Postgraduate School, unpublished Excel spreadsheet.

[8] M. R. Driels, “Indirect Fire Weapon Effectiveness Computation,” Department of Mechanical Engineering, Naval Postgraduate School, unpublished Excel spreadsheet.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California